SUPER 6

Anno 2
n. 6 - Giugno 1985

L. 7.500

Spedizione in abbonamento postale Gruppo III/70

COMMODORE

SUPERCOMMODORE
la rivista per gli utenti dei prodotti COMMODORE

Un pubblicazione della
J.soft EDITRICE
in collaborazione con



GRUPPO EDITORIALE JACKSON

SUPERGRAPHIC PER C64

UN PICCOLO DATABASE

ASTROPANIC PER VIC 20

FORBIDDEN CRYPT

CYPHER

PERSONAL

ATARI 130XE

NUMERI, NON PAROLE.

ATARI 130 XE è un grande personal computer che, ad una enorme capacità di memoria, abbina una maggiore versatilità. Atari 130XE non ha bisogno di molte parole per essere presentato, i suoi numeri parlano da soli.

128K RAM

ATARI 130XE utilizza un microprocessore 6502C.
Ha una memoria di 128K RAM, 24K ROM (Sistema operativo e linguaggio programmazione Basic). Tanta potenza in più per un maggior numero di informazioni.

4 CHIPS SPECIALI

Sono Chips esclusivi Atari: GTIA - Per la visualizzazione grafica; POKEY - Generatore e controllo di suono; ANTIC - Per la gestione dello schermo e di Input /Output; FREDDY - Sistema di controllo della memoria.

256 COLORI

ATARI 130XE ha capacità grafiche nettamente superiori alla media: 11 tipi di modi grafici; 256 colori (128 visualizzati contemporaneamente); grafica 320 x 192 ad alta risoluzione;
una visualizzazione di 40 colonne per 24 linee; 5 modi di testo. Tutto quanto ti serve per scatenare al meglio la tua creatività, senza limiti di alcun tipo.

4 CANALI SONORI

Anche per le capacità sonore, Atari 130XE rivela straordinarie e inaspettate possibilità: 4 canali sonori indipendenti su una gamma di 3 ottave e mezza.

62 TASTI

Una tastiera formidabile: 62 tasti, tasto Help e quattro tasti per funzioni speciali; caratteri internazionali; 29 tasti con funzione grafica. Linguaggio di programmazione Atari Basic incorporato.

41 CONFIGURAZIONI POSSIBILI

Software e periferiche compatibili con i computers della serie XL: Stampante a colori Atari 1020; stampante "Letter Quality" Atari 1027; Stampante ad aghi programmabile Atari 1029; Disk drive Atari 1050 (È possibile l'utilizzo di 4 unità in serie); Joystick e Super Controller.

L. 380.000 + IVA



ATARI® COMPUTERS

TECNOLOGIA FORTE, PREZZO VINCENTE.

I NOSTRI RIVENDITORI: Lazio Videosuono V.A. Baldovinetti 68/74 - Roma 06/5038525-5035880 - **Lombardia + Novara** Consolo & Longoni V.le dell'Industria 63 - Paderno Dugnano 02/9183372-9184083 - **Piemonte - Valle d'Aosta** Norvat C.so Fiume 12 - Torino 011/682171 - **Tre Venezie** Interservice V.S. Pietro 58/A - Padova 049/655654 - **Emilia+S. Marino** Telcom Via dei Mille 19 - Bologna 051/274600 - **Toscana, Umbria, Marche+La Spezia** Telebit c/o Sig. Cattani Via Il Prato 8/R - Firenze 055/262652 - **Puglia, Abruzzo, Basilicata, Molise** Domina V. Turati 16 - Bari 080/420204 - **Sicilia, Calabria** Belco V. Mariano d'Amelio 78 - Palermo 091/547566 - **Sardegna** Sicom Strada Sestu - Elmas Km. 2,1 Sestu - Cagliari 070/22317 - **Campania** Lada V. Ferrante Imparato 33 S.Giovanni a Teduccio (NA) 081/7527006
DISTRIBUZIONE NAZIONALE Canale Foto Cine Circe Film Via Pascoli 70/3 - Milano 02/2363556 - 2366410

Anno 2 · n° 6 · Giugno 1985

SUPER COMMODORE

J. soft s.r.l.
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Mauro Cristuib Grizzi

HANNO COLLABORATO A QUESTO NUMERO
U.G. Barzaghi
A. Motta
F. Sarcina

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70

Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500 (11 numeri con cassetta);
L. 38.500 (11 numeri senza cassetta);
per l'estero: L. 110.000 (11 numeri con cassetta) - L. 50.000 (11 numeri senza cassetta)

I versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5 - 20124 MILANO
mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c postale n. 19445204.
Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA COMPUTE! O DA COMPUTE! GAZZETTE - COMPUTE! PUBLICATION, INC.
P.O. BOX 5406,
GREENSBORO, NC 27403 - USA

La biblioteca per ragazzi firmata Jackson.

Daniel Isaaman - Jenny Tyler

BATTAGLIE CON IL COMPUTER

Il gioco della guerra ha contagiato anche i computer. Qui vengono presentati vari giochi scritti in BASIC per "fare la guerra" sul proprio computer, che deve essere uno dei seguenti: BBC, Apple, Vic 20, TRS-80, ZX Spectrum e ZX 81.

Cod. 011D Pag. 48 Lire 9.000

Ian Graham

GIOCHI CON IL COMPUTER

Il libro, parla di giochi con il computer, una volta tanto visti dalla parte del computer, e non dell'utente.
In particolare spiega, in modo semplice ma preciso, come fa il computer a giocare, come fa a produrre immagini e suoni durante il gioco e come fa (in genere!) a vincere.

Cod. 006D Pag. 48 Lire 9.000

Brian Reffin Smith

IMPARIAMO A PROGRAMMARE

Il libro fornisce le conoscenze essenziali per incominciare a programmare in BASIC su di un personal computer.

Cod. 018D Pag. 48 Lire 9.000

PRIMI PASSI IN BASIC

Il libro propone l'apprendimento del linguaggio BASIC tramite lo studio e l'esame dettagliato di programmi già scritti.

Cod. 007D Pag. 48 Lire 9.000

Tony Potter - Ivor Guild

I ROBOT

Dopo una rapida panoramica su che cosa può fare un robot, il libro presenta una serie di robot con funzioni particolari: i robot a braccio, i robot operai, i robot nello spazio, etc.
Affronta poi il problema di come un robot può essere programmato e controllato da un computer, e di come praticamente si realizza un robot.

Cod. 003D Pag. 48 Lire 9.000

Judy Tatchell - Bill Bennett

CONOSCERE IL PERSONAL

Il libro spiega che cosa si può fare con un personal computer, come si usa e come funziona.

Cod. 008D Pag. 48 Lire 9.000

Lynn Miring - Ian Graham

RIVOLUZIONE INFORMATICA

Il volume è rivolto in particolare ai bambini, ma anche a tutti coloro che, presto o tardi, verranno coinvolti dalle nuove tecniche informatiche più come utenti che come operatori.

Cod. 004D Pag. 48 Lire 9.000



GRUPPO EDITORIALE JACKSON

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ________ n° ________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome ________

Cognome ________

Via ________

Cap. ____ Città ________ Prov. ____

Data ____ Firma ________

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. | | | | | | | | | | | |

ORDINE MINIMO L. 50.000

POINT

L'importante è non esagerare

D'accordo, c'è la passione, c'è la voglia di fare e imparare sempre cose nuove, ma se la cosa diventa un fatto "patologico", questo non va bene. Di giorno ci sono le attività a cui ciascuno di noi deve necessariamente sottendere, la notte, salvo eccezioni (leggi guardiani notturni), è necessario dormire.
Dunque, il tempo da dedicare al computer dev'essere recuperato dal cosiddetto "tempo libero" che, per quanto sopra, subentra la costrizione di dover operare sul calcolatore solo per brevi periodi di tempo, che a volte condizionano la riuscita di quanto si desidera fare.
Quando si pone questa condizione gli utenti solitamente si trasformano in una sorta di "zombies", che passano ogni ora disponibile (comprese quelle notturne) appiccicati alla tastiera, cercando di far digerire alla macchina quanti più byte possono.
Se questo modo di agire diviene prassi ricorrente, le implicazioni sono quasi sempre poco piacevoli. Bocciature, liti furibonde con il coniuge, insonnia cronica e via dicendo.
La sostanza di tutto il discorso si riassume in un sol motto: non esagerare! Senza contare poi che, di solito, quando ci si accanisce con un programma che non vuole girare a dovere, è molto difficile porvi rimedio se la mente non è riposata.
L'importante è non lasciarsi coinvolgere al punto da trasformare il computer nell'unico interesse della propria vita, dedicandogli pertanto tutte le risorse di tempo e di energia di cui l'individuo dispone.
Il consiglio dunque è uno solo: misura e moderazione che, oltre ad evitare spiacevoli inconvenienti, faranno maggiormente apprezzare il tempo, anche se poco, speso sulla tastiera del computer a dare legittimo sfogo alla propria passione.

Pietro Dell'Orco

Per collaborare a SUPERCOMMODORE

La rivista è interessata ad articoli e programmi riguardanti la linea Commodore, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Sprite Ed.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente.
È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare le istruzioni "CHR$(x)", "TAB(x)", SPC(x)", piuttosto che stringhe di manipolazione del cursore.
Ad esempio, per uno scroll di 5 linee l'istruzione "FORI = 1TO5:PRINT:NEXT". è molto più interpretabile di 5 Q inverse; e, invece di una dozzina di simboli di cursore a destra, perché non usare semplicemente "PRINT SPC (12)"? Un rapido controllo dei programmi per operare queste sostituzioni sarà molto apprezzato da noi e dai lettori.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24×36, o 6×6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

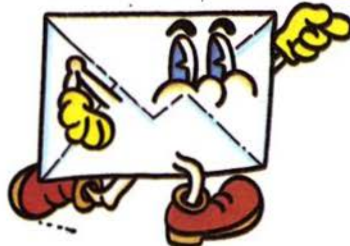
15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviate idee e programmi a:
SUPERCOMMODORE
Viale Restelli, 5
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

READ &WRITE



La posta dei lettori

Warm Start su VIC e C64

Leggendo la risposta ad una lettera su un numero di SUPERCOMMODORE, ho scoperto che si può riportare il C64 alle condizioni di "appena acceso" con la SYS 64738. Tale SYS cancella però totalmente il contenuto della memoria del computer: esiste una SYS che simuli invece la pressione di RUN/STOP + RESTORE?

Mario Colombo

R. Ritorniamo sull'argomento "cold start e warm start", anche per rispondere ad altri quesiti posti da lettori, che, malgrado la SYS 64738, si ritrovavano ancora in memoria programmi in linguaggio macchina residenti a partire da 49152.
Con una SYS 64738 si entra nella routine di sistema operativo di "cold start".
Questa routine è la prima ad essere eseguita automaticamente quando si accende il computer, ed ha lo scopo di resettare allo standard tutti i puntatori di memoria, porre i colori di schermo ai valori di default, regolare il set di caratteri sul modo maiuscolo/grafico, porre allo standard tutti i vettori dell'I/O e controllare se sia inserita qualche cartridge. L'unica differenza tra l'effetto di questa SYS ed il reale spegnimento e riaccensione del computer è lo "shock" dovuto al transitorio di corrente in quest'ultimo caso.
I programmi in linguaggio macchina che risiedono a partire da 49152 vengono cancellati, se si spegne e riaccende il computer, a causa di questo transitorio, che, provocando il passaggio dei valori di corrente e di tensione da zero al valore prescritto, "scompiglia" il contenuto dei byte. Un esperimento: provate ad esempio a caricare in memoria – poniamo, al solito, a partire da 8192 – una pagina grafica; quindi spegnete e riaccendete il computer, senza ovviamente lasciar passare troppo tempo tra un'operazione e l'altra, e date le POKE di attivazione della bitmap a partire da 8192.
Teoricamente dovreste vedere solo il caos, mentre potrete notare che qualcosa – e forse più di qualcosa – è rimasto della precedente pagina grafica. Ciò avviene perché – anche se non in tutta la RAM – la corrente e la tensione impiegano un certo tempo ad azzerarsi in corrispondenza dello spegnimento del computer, e può quindi capitare che riaccendendo la macchina si possano trovare zone di RAM in cui corrente e tensione non si sono discostate troppo dal valore precedente, conservando quindi i dati anche se in maniera distorta e non più utilizzabile.
La pagina grafica è un mezzo ideale per rendersi conto di questo fenomeno, in quanto ciò che vi è visualizzato è facilmente riconoscibile. La stessa cosa capita naturalmente anche ai programmi, tuttavia per questi ultimi sappiamo bene che è sufficiente... un byte fuori posto per rendere il tutto irriconoscibile ed indistinguibile dalla RAM con il contenuto "veramente" casuale.
La SYS64738, ovviamente, non provvede anche a "scompigliare" i byte in modo casuale, e quindi è più che naturale ritrovare il linguaggio macchina a partire da 49152. Ciò è certamente un vantaggio, perché non presenta comunque alcun tipo di inconveniente.
Il "warm start", o inizializzazione a caldo del computer, corrisponde alla pressione contemporanea dei tasti RUN/STOP e RESTORE. Poiché il tasto RESTORE è collegato direttamente al microprocessore ed invia un segnale NMI (Non Maskable Interrupt = interruzione non mascherabile) ogni volta che esso viene premuto, non esiste una vera e propria routine di sistema operativo che esegue queste funzioni. È tuttavia possibile, dal momento che la funzione eseguita da RUN/STOP+RESTORE è equivalente all'esecuzione di un'istruzione BRK in linguaggio macchina, fare in modo che questa istruzione venga eseguita.
Il valore di BRK è 0, quindi basterà "pokare" uno zero in una locazione di memoria qualunque e disponibile, quindi fare una SYS a questa locazione, ad esempio, POKE 2048,0:SYS2048 (per il C64), oppure POKE828,0:SYS848 (per il VIC 20 con o senza espansioni).

Il modo colore di fondo esteso

Tra i vari modi grafici disponibili sul mio C64, ho letto che esiste anche un "modo colore di fondo esteso", ma non mi è chiaro in che cosa consista. Potete darmi alcune delucidazioni?

Ignazio Rocca

R. Il modo colore di fondo esteso è un modo grafico appartenente alla famiglia dei sistemi di modifica dei caratteri. Esso controlla il colore principale del carattere (foreground) ed il suo colore di fondo (background) in modo autonomo dal colore di fondo su cui è impostato lo schermo. Con questo sistema è possibile ottenere, ad esempio, uno schermo nero con un carattere verde a sfondo bianco. La limitazione nell'utilizzo di questo "modo" consiste nella disponibilità dei soli primi 64 caratteri del set, cioè dalla "@" al "?". Ciò avviene dal momento che vengono usati per la definizione del colore di fondo del carattere due bit del suo codice schermo. Esistono quattro registri colore, ognuno dei quali può essere ovviamente impostato con uno dei 16 colori disponibili nel C64, mentre la memoria colore conterrà il colore di fondo del singolo carattere.
Per accedere al modo colore di fondo esteso occorre porre a 1 il sesto bit della locazione 53265. Ciò si ottiene con la seguente istruzione:
POKE53265,PEEK(53265)OR64

La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione.
Un computer possiede potenzialità, ma, come una macchina senza benzina, senza un programma non può funzionare.
La maggior parte dei programmi pubblicati su **SUPERCOMMODORE** per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese **SUPERCOMMODORE** pubblica programmi sia per il VIC 20 che per il C64. Tanto per cominciare, se possiedi un VIC 20 copia solo i programmi scritti per la tua macchina, indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire programmi scritti per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa.
Ogni lettera, carattere o numero ha il suo significato.
Un errore banale è costituito dalla sostituzione del numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere maiuscolo "B" con il numero "8".
Devi anche inserire tutta la punteggiatura, i due punti (:) e le virgole, copiando esattamente ciò che appare sulla rivista. Gli spazi possono essere importanti. Per essere sicuro **copia il listato esattamente** come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerai indicazioni tra parentesi, quali: "{GIÙ}".
Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fai riferimento alla "Piccola guida per l'input dei programmi".

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le informazioni di cui il programma ha bisogno.
In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, in altri contengono codici grafici. Queste linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato, il calcolatore potrebbe "piantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.
Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria, per cui è sempre necessario fare il SAVE del tuo programma prima di comandare il RUN.
Se il computer si ferma, puoi caricare (LOAD) il programma e cercare l'errore.
A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.
Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.
Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma, se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inverse", i caratteri minuscoli e quelli di controllo?
Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

1) Copia il programma una linea alla volta, con ordine. Premi RETURN alla fine di ogni linea. Usa il tasto "DEL" per correggere gli errori.
2) Confronta la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presenti un errore quando esegui il RUN.
3) Accertati di aver inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input dei programmi" che trovi in questa stessa rivista).

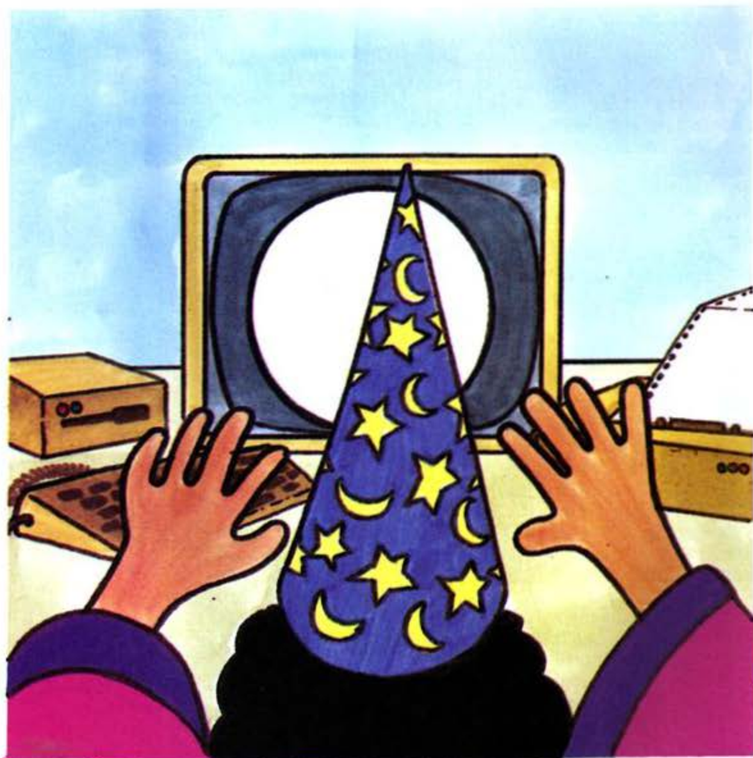
Siamo spiacenti di non poter rispondere singolarmente alle richieste di informazioni circa i programmi, prodotti o servizi apparsi su SUPERCOMMODORE.



di **M. Cristuib Grizzi**

Il (prossimo) futuro dei sistemi periferici

Nel settore degli "home computer" c'è stata, negli ultimi anni, una continua e progressiva evoluzione, che ha permesso di passare in poco tempo da macchine limitate come lo ZX80 ad altre più potenti e sofisticate, come il C64 od il recente MSX. Non si è invece verificata un'analoga evoluzione tecnologica nei sistemi di memoria di massa, che, al contrario, sono rimasti sostanzialmente quelli che già esistevano quattro o cinque anni fa.



La velocità di calcolo del computer ci permette di ottenere prestazioni matematiche e di gestione dati che, fino a qualche tempo fa, richiedevano invece un duro e lungo lavoro, accompagnato da carta e penna; non sempre tuttavia il sistema informatico di cui disponiamo si rivela la miglior soluzione per trattare un certo problema.

Prendiamo in considerazione, ad esempio, uno degli utilizzi "pratici" del computer che per primi vengono in mente al novello utilizzatore: un'agenda telefonica completamente gestita dal computer.

Supponiamo che si sia entrati in possesso, prendendolo da una rivista o acquistandolo, di un programma di archivio indirizzi tale da soddisfare le nostre – limitate – esigenze di immagazzinamento di una serie di nomi, indirizzi e numeri di telefono.

Dopo aver compiuto il noioso, ma inevitabile lavoro di inserimento nel computer di tutti questi dati, potremo finalmente vantarci con gli amici di poter ricercare il numero di telefono e l'indirizzo del signor Rossi, semplicemente digitando sul nostro computer i caratteri "Ros" e premendo RETURN. Nello spazio di pochi attimi avremo visualizzato sul video, o stampato su carta, ciò che ci serve e – perché no – potremo avere un elenco in ordine alfabetico del contenuto dell'agenda, oppure potremmo fare in modo che certi numeri di telefono riservati vengano visualizzati solo dopo aver inserito una certa parola chiave.

Tutto vero, ma forse la cosa più comoda è ancora usare la vecchia e cara agenda, con i nomi cancellati, le scritte disordinate e di colore diverso: e la possiamo consultare con un tempo di accesso medio di qualche secondo...

Il programma di archiviazione telefonica, infatti, risulta comodissimo se da mane a sera possiamo tenere impegnato il computer su di esso: in questo caso, tuttavia, disporremmo di una rubrica telefonica piuttosto cara...

Se, al contrario, quando ci serve un numero telefonico dobbiamo spegnere il computer per uscire dal word processor che stiamo utilizzando, cercare la cassetta con il programma di archiviazione telefonica, attendere che esso venga caricato in memoria, attivarlo e quindi attendere ancora che vengano caricati tutti i dati registrati su nastro, vediamo facilmente che per cercare un numero di telefono avremo un tempo di accesso medio di una decina di minuti. Allora, a che serve la velocità di calcolo del computer?

Non è il computer a rivestire la parte dell'imputato, ma è il sistema di memorizzazione di massa a non reggere il passo con le prestazioni ottenibili dal computer stesso.

Il sistema di registrazione Commodore, sia di file binari (programmi) che di file ASCII (dati), risale ai tempi gloriosi delle prime serie PET, ed è stato ancora mantenuto senza modifiche anche per i nuovi modelli C16 e Plus/4. Non molto dissimile è la situazione del sistema di memorizzazione dati su floppy disk: per chi passa dall'abitudine a lavorare con cassetta ad un sistema a disco sembra di aver sfondato la barriera della velocità della luce, ma ci si abitua velocemente e non si tarda a rendersi conto di quanto anche il caricamento da disco avvenga con esasperante lentezza rispetto ad altri sistemi.

Fortunatamente, anche il settore degli apparati di memoria di massa si sta svegliando dalla lunga stasi e promette, a medio se non a breve termine, innovativi e sostanziali cambiamenti.

I "compact disk", già entrati nel mercato audio ed HI-FI da qualche tempo, rappresenteranno – e rappresentano già, in alcuni sporadici casi – un interessante strumento per la memorizzazione di massa anche per computer: attualmente i "compact disk" per computer possono essere soltanto letti, in quanto per la loro registrazione sono richieste complicate apparecchiature dai costi proibitivi. Malgrado questa limitazione, il loro punto di interesse sta nella enorme massa di dati immagazzinabile su una facciata di questi dischi: la Nippon Columbia ha prodotto un tipo di "compact disk" dedicato al computer, che può immagazzinare fino a 550 Mbyte per facciata (equivale a dire che un singolo disco a doppia faccia contiene tutti i dati ora immagazzinati su un insieme di circa 6400 floppy disk per 1541!).

Con una misura del diametro di circa cinque pollici – più o meno come gli attuali floppy – i "compact disk" sfruttano lo stesso meccanismo di lettura laser dei "cugini" audio, e sono attualmente sottoposti a studi e ricerche per realizzare un sistema a basso costo, che permetta di ottenere anche la possibilità di registrazione dati.

Un altro sistema – anche se un po' particolare – di memoria di massa che sarà presto disponibile è il videodisco. Sono già comparsi anche in Italia i primi videogiochi "da bar" che lo utilizzano, dotati di dischi sui quali è memorizzato un gran numero di immagini digitalizzate, tali da permettere al giocatore di manovrare un personaggio od un oggetto in un cartone animato – od addirittura in un film – totalmente interattivo.

Questi dischi sono in grado di contenere fino a 50000 immagini o fotogrammi diversi, con un tempo medio di accesso rapidissimo. La Digital Research ha già commercializzato sul mercato statunitense una particolare interfaccia del costo – limitatissimo – di 49 dollari, che permette di collegare il C64 ad un lettore di videodischi.

I sistemi di memorizzazione a nastro magnetico, già oggi arcaici anche se – ed in parte proprio per questo – più economici, sono destinati a non avere ancora vita lunga: appena sarà sviluppata una tecnologia di produzione dei nuovi sistemi economicamente competitiva, saremo finalmente posti in condizioni tali da poter sfruttare appieno la velocità di calcolo insita nel nostro computer.

Certamente lo sforzo profuso nella ricerca dalle grosse aziende produttrici di hardware non si limita agli apparati di memorizzazione di massa, ma è orientato anche verso una nuova serie di periferiche che, immaginate ora, possono avere del fantascientifico. In realtà, sono solo dietro l'angolo.

La Synapse Corporation (nota senz'altro a chi si interessa di videogame), già vende in USA un joystick completamente... rivoluzionario: il "Relax" – questo è il suo nome – consiste in un leggero sensore che si sistema sul capo del giocatore e, tramite impulsi elettromagnetici (piccolissimi impulsi generati dal lieve movimento dei muscoli del capo), permette di controllare lo svolgimento di un videogioco o comunque di un programma, comodamente seduti in poltrona e senza toccare fisicamente né tastiera né joystick "tradizionali" di alcun tipo.

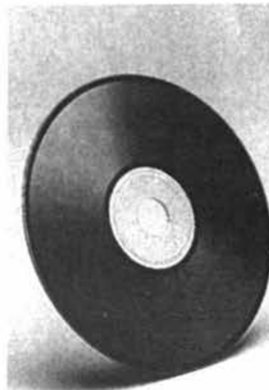
Sempre in USA viene già venduto un accessorio per l'Apple Macintosh, in grado di "leggere" i testi che vengono fatti scorrere sopra il rullo della stampante ed immagazzinarli in memoria centrale o su disco: è la fine ufficiale del lavoro di digitazione di listati o di copiatura di qualsiasi testo!

Gli schermi video a cristalli liquidi a colori (LCD: liquid crystal display), con la loro altissima risoluzione, stanno per uscire dallo stadio dei prototipi: mentre gli schermi a cristalli liquidi monocromatici sono ormai da tempo usati per computer, calcolatrici ed altre applicazioni, l'aggiunta del colore sembrava fino a ieri un problema di difficile soluzione. Ora, con i transistor al silicio a film sottile, il colore sta iniziando a comparire su schermi di dimensioni ridotte, e presto sarà disponibile anche su schermi di grosse dimensioni.

Infine, malgrado la necessità dei modem sia da noi poco sentita a causa della ancora non totale diffusione dei personal e di alcune scelte politiche, è interessante apprendere che in USA è stato presentato il primo modem via radio, in grado di ricevere programmi da un normale apparecchio radio a modulazione di frequenza alla velocità di ben 4800 baud e del costo di soli 70 dollari.

Come si vede, il mondo che gravita intorno al computer è in piena "effervescenza" e sarà in grado sul breve termine di offrirci novità ed apparecchiature che oggi possono ancora sembrare uscite dalle pagine di un moderno Jules Verne.

Il "Compact disk", il cui diametro misura circa cinque pollici, può diventare un supporto di memoria di massa per i personal ed home computer.



I dischi laser possono contenere su ogni lato fino a 50.000 immagini complesse differenti.

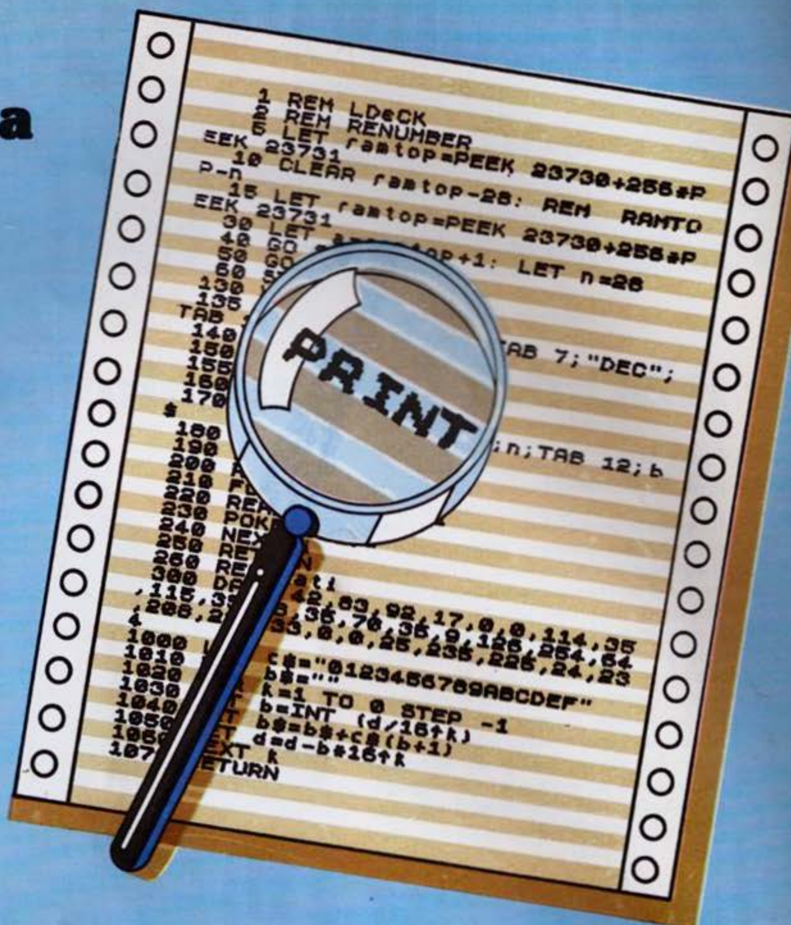


Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERCOMMODORE** contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, ecc.).
Per rendere chiaro ciò che dev'essere battuto quando è necessario inserire uno dei suddetti caratteri sono state stabilite le seguenti convenzioni.
Generalmente i listati per VIC 20 e C64 contengono delle «parole» racchiuse tra parentesi graffe { }; tali parole rappresentano particolari caratteri di controllo: {**GIU**} significa premere il simbolo del cursore verso il basso. {**5 SPAZI**} vuol dire battere 5 volte la barra-spazio.
Per indicare che un tasto dev'essere «shiftato», cioè premuto insieme al tasto SHIFT, nel listato il simbolo di questo tasto sarà sottolineato.
Per esempio S significa che dev'essere premuto il tasto S mentre è abbassato il tasto SHIFT.
La manovra farà apparire sullo schermo un piccolo «cuore».
Trovando un simbolo sottolineato chiuso tra parentesi graffe (esempio {**10 N**}) dovrà essere interpretato come «premere il carattere indicato per il numero di volte che lo precede nelle parentesi»: nel nostro esempio premere 10 volte il tasto N «shiftato».
Se il tasto da premere è indicato tra le parentesi [⟨⟩], vuol dire che lo stesso dovrà essere premuto mentre è premuto il tasto «Commodore» (il tasto «Commodore» è quello posto nell'angolo in basso a sinistra).
Se il tasto scritto tra [⟨⟩] è preceduto da un numero, ciò significa che il tasto dev'essere premuto per il numero di volte indicato.
Raramente si potrà trovare un carattere alfabetico solitario racchiuso tra parentesi graffe. Tale carattere sul C64 può essere battuto mentre è premuto il tasto CTRL.
Ad esempio {**A**} sta ad indicare la sequenza CTRL-A.
A proposito del «modo virgolette» è nota la possibilità di muovere il cursore sullo schermo con i tasti CRSR.
Ogni tanto i programmatori desiderano muovere il cursore durante l'esecuzione del programma. È per questo motivo che nei programmi si troveranno dei {**SIN**}, {**HOME**} e {**BLU**}.
L'unico modo perché il calcolatore distingua il comando diretto da quello inserito nel programma è il «modo virgolette». Premendo il tasto «virgolette» (tasti SHIFT 2) il calcolatore si predispone appunto in «modo virgolette».
Battendo un carattere qualsiasi, cercando poi di correggerlo muovendo il cursore, si otterrà solo un tratto verticale in un quadratino inverso. Questo è infatti il simbolo del cursore a sinistra nel «modo virgolette».
L'unico comando di editing non utilizzabile all'interno di un programma è il DEL. Battendo nuovamente le «virgolette» il calcolatore lascia il «modo virgolette». Si può accedere al «modo virgolette» quando inserite spazi nella linea.
In ogni caso, il sistema più semplice per uscire dal «modo virgolette» è quello di premere il tasto RETURN.
Utilizzare la tabella che segue quando è necessario inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	C= 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	C= 8	
{SU}	SHIFT ↑CRSR↓		{GRN}	CTRL 6		{F1}	f1	
{GIU'}	↑CRSR↓		{BLU}	CTRL 7		{F2}	f2	
{SIN}	SHIFT ⇐CRSR⇒		{YEL}	CTRL 8		{F3}	f3	
{DES}	⇐CRSR⇒		[<1>]	C= 1		{F4}	f4	
{RVS}	CTRL 9		[<2>]	C= 2		{F5}	f5	
{OFF}	CTRL 0		[<3>]	C= 3		{F6}	f6	
{BLK}	CTRL 1		[<4>]	C= 4		{F7}	f7	
{WHT}	CTRL 2		[<5>]	C= 5		{F8}	f8	
{RED}	CTRL 3		[<6>]	C= 6				

OK-List per la perfetta battitura dei listati



Inserite in memoria l'OK-LIST prima di accingervi a battere i listati pubblicati: avrete a disposizione un formidabile mezzo per essere sicuri di aver inserito nel computer ogni carattere in modo assolutamente corretto.

Tutti i nostri listati sono caratterizzati da un numero aggiunto ad ogni linea di programma, ad esempio: rem 123. NON COPIATE QUESTA PARTE DELLA LINEA! Serve unicamente per vostra informazione. L'istruzione REM ha l'unico scopo di renderla inoffensiva, se per errore doveste batterla.

Se, prima di iniziare il lavoro di battitura del listato, caricate in memoria OK-LIST e lo attivate con RUN, per ogni linea che battete verrà visualizzato un numero (detto checksum) nell'angolo superiore sinistro dello schermo.

Confrontate questo numero con quello pubblicato per ogni linea del listato: se non corrisponde, ciò significa che avete commesso un errore nella battitura della linea stessa.

Le ore buttate per cercare e correggere errori di battitura in programmi che non funzionano saranno solo un lontano ricordo!

Due ultime cose di cui tenere conto:

OK-LIST non tiene conto degli spazi: questo per vostra convenienza, poiché questi generalmente sono poco importanti.

Infine, OK-LIST è allocato nel buffer del registratore (locazioni 886-1018), quindi, prima di salvare su cassetta il programma che state battendo, ricordatevi di disabitarlo premendo contemporaneamente RUN/STOP + RESTORE. Potrete poi riattivarlo con SYS 886.

Il sistema più comodo per usare OK-LIST è quello di battere normalmente il programma, quindi listarlo e portarsi con il cursore sul numero di linea più basso che appare sul video; premere quindi RETURN e confrontare il numero di check-sum visualizzato nell'angolo superiore sinistro del video con quello pubblicato per la data riga del programma. Il cursore si porterà automaticamente sulla linea successiva e non dovrete fare altro che premere il checksum seguente.

Data l'area di memoria in cui si trova, OK-LIST non può essere usato per controllare un programma già salvato su cassetta, mentre ciò è possibile se il programma è stato registrato su disco.

OK-LIST versione per VIC 20 e C 64

```
100 PRINT"{CLR}ATTENDERE PREGO..."
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:
    NEXT
110 IF CK<>17539 THEN PRINT"{GIU'}ERRORE
    NELLE ISTRUZIONI DATA":END
120 SYS886:PRINT"{CLR}{ 2 GIU'}OK-LIST AT
    TIVATO.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```



ABC del Linguaggio macchina

Parte terza
di **U. G. Barzaghi**

In questa penultima parte del nostro breve viaggio attraverso il linguaggio macchina ci occuperemo del trasferimento di blocchi di dati, affrontando eventuali argomenti non ancora trattati e in particolare completando, in maniera più organica di quanto fatto nella puntata precedente, l'argomento riguardante le tecniche di indirizzamento.
Introdurremo, come di consueto, alcuni programmi esemplificativi, tra cui una semplice routine di "scrolling".



Qual è l'utilità di una routine in linguaggio macchina per il trasferimento di blocchi di dati? Coloro, tra voi, che possono vantare una più appofondita conoscenza del linguaggio macchina sanno che simili routine sono parte integrante di qualsiasi programma assemblatore degno di questo nome. Anche il nostro "Supermonitor", infatti, dispone della funzione "T", che consente il trasferimento di locazioni di memoria.
Tale istruzione richiede di specificare l'indirizzo iniziale e finale del blocco di dati che si desidera trasferire e l'indirizzo iniziale della sua nuova posizione. Il trasferimento avviene a partire dall'ultima locazione del blocco, il che consente, nel caso in cui le due aree siano in parte sovrapposte ed il trasferimento avvenga nel senso crescente degli indirizzi, di non cancellare il contenuto del blocco che si desidera trasferire, prima che il trasferimento sia avvenuto. Ovviamente, nel caso in cui lo spostamento avvenga in senso opposto e le due aree si sovrappongano il risultato non sarebbe corretto e sarà quindi necessario ricorrere ad una doppia operazione, utilizzando un'area di memoria libera come memoria "tampone".
Sarà necessario tener conto di questi particolari nello sviluppare i nostri sottoprogrammi in linguaggio "ASSEMBLY" onde evitare spiacevoli inconvenienti.

Perché dunque realizzare programmi in linguaggio macchina per il trasferimento di blocchi, quando questi possono essere opportunamente posizionati tramite "Supermonitor" e quindi caricati in memoria, direttamente nella posizione originale, da nastro o da disco?
Le ragioni sono molteplici. Ad esempio perché determinate aree della memoria non sono accessibili nemmeno per la versione rilocata del nostro assemblatore.
Abbiamo visto nel numero scorso come liberare l'area di memoria che va da 49152 ($C000) a 53248 ($D000), trasferendo il codice ASSEMBLY del programma assemblatore a partire da 8192 ($2000). In questo modo rimane però invariata la posizione del sistema operativo da 57344 a 65535 ($E000 - $FFFF), delle cui routine il programma fa uso e la rilocazione integrale del quale sarebbe, quanto meno, problematica.
La zona occupata dal sistema operativo è interdetta al programma assemblatore e qualsiasi tentativo di trasferire dati in quelle locazioni avrà come risultato o una operazione di trasferimento incompleta o, addirittura, il blocco critico del sistema, che vi costringerà a spegnere il calcolatore, perdendo eventuali dati precedentemente inseriti.
Si può, ovviamente, rinunciare ad usare

quelle locazioni di memoria.
Esse sono, però, particolarmente comode. Trasferendo infatti il blocco di memoria di 16Kbyte "visto" dal chip VIC-II di gestione del monitor, a partire dalla locazione di indirizzo 49152 (Banco 3), si possono inserire in questo banco di memoria fino a quattro blocchi di memoria di schermo o due set completi di caratteri ridefiniti, ponendoli al di sotto dei chip video (6566; da 53248 a 53294) e audio (6581 SID, Sound Interface Device, dispositivo di interfacciamento sonoro; da 54272 a 54300). Nello stesso banco, al di sopra della memoria colore (da 55296 a 56319) e dei chip di interfaccia 1 e 2 (IRQ, Interrupt ReQuest, richiesta di interruzioni, da 56320 a 56335 e NMI, Not Maskable Interrupt, interruzioni non mascherabili, da 56576 a 56591), possono trovare posto otto schermi a bassa risoluzione od uno schermo intero ad alta risoluzione, oppure fino a quattro set di caratteri personalizzati. Operando opportunamente, è possibile archiviare in questa zona, che si trova completamente al sicuro da eventuali interferenze da parte di programmi BASIC (dal momento che l'interprete BASIC risiede alle locazioni da 40960 a 49151), tutto quanto può servire alla gestione della grafica di programmi BASIC anche molto complicati. Nel caso di uno degli ultimi programmi da noi realizzati per la J.*soft* (la versione per C64 della "Battaglia di Abukir") siamo riusciti ad inserirvi due schermate a bassa risoluzione ed uno schermo colore ausiliario, un set di caratteri personalizzati completo e la forma di ben 96 sprite diversi; oltre ad avere ancora spazio per i sottoprogrammi in linguaggio macchina necessari alla gestione del tutto, proprio quei programmi che illustreremo in questo articolo.
Un primo scopo delle nostre routine consisterà quindi nel consentire il caricamento rapido dei nostri dati, utilizzando una zona di memoria ausiliaria, che potremo quindi "rilasciare" ad eventuali programmi BASIC, dopo aver provveduto ad un trasferimento rapido dei dati nella locazione di memoria voluta. Vale la pena di affrontare l'argomento in forma compiuta, per chiarire anche eventuali dubbi che vi possano essere sorti riguardo l'utilizzo dei programmi registrati tramite "Supermonitor".
Questi possono essere caricati in memoria direttamente alle locazioni da cui provengono, a patto di specificare, tramite il suffisso ', 1' apposto alla istruzione di caricamento, che non si tratta di programmi BASIC, ma ASSEMBLY o di blocchi di dati.
Supponendo quindi di aver registrato una subroutine in linguaggio macchina tramite l'istruzione

```
.S 8000 9FFF "PROVA" 08
```

sarà possibile ricaricarla nelle locazioni di provenienza (32768-40959), senza aver bisogno del programma assemblatore, tramite l'istruzione

```
LOAD "PROVA",8,1
```

Tutto questo nel caso in cui si utilizzi come periferica l'unità dischi; nel caso di registrazione su nastro non è indispensabile specificare la periferica per la prima istruzione, che diverrà quindi

```
.S 8000 9FFF "PROVA"
```

o, anche

```
.S 8000 9FFF "PROVA" 01
```

mentre nel secondo caso è necessario specificare il numero che identifica la periferica utilizzata tramite l'istruzione

```
LOAD "PROVA",1,1
```

Queste istruzioni consentono di caricare, come detto, il nostro programma in linguaggio macchina, o i dati di cui desideriamo disporre, direttamente nella zona di memoria loro destinata (tentativi di interferire con la pagina zero – utilizzata, fatta eccezione per le locazioni specificate nel numero scorso, dal sistema operativo – o con l'interprete BASIC provocheranno, ovviamente, malfunzionamenti del programma o dati inutilizzabili).
Esse sono, inoltre, altrettanto rapide del metodo visto nelle puntate precedenti e consistente nell'inserire nelle locazioni di memoria volute i valori decimali corrispondenti alla codifica ASSEMBLY del nostro programma, leggendoli da frasi DATA. Ciò consente di risparmiare lo spazio in memoria che risulterebbe altrimenti necessario per ospitare il contenuto delle frasi DATA (che, data la codifica delle istruzioni DATA in BASIC, comporta l'utilizzo di una zona di memoria ben maggiore di quella effettivamente utilizzata dal nostro programma), frasi che risultano del tutto inutili una volta che il loro contenuto è stato letto.
Una semplice alternativa è rappresentata dalla creazione di file sequenziali su disco, da cui leggere i dati con un ciclo di inserimento in memoria, non dissimile da quello utilizzato per le frasi DATA. Questo sistema, però, ha il non piccolo inconveniente di essere, anche su disco, mortalmente lento (come abbiamo avuto modo di sperimentare personalmente) e non osiamo pensare a quali sarebbero i suoi effetti su nastro.
Ecco quindi che il caricamento tramite le istruzioni suddette risulta particolarmente attraente. Il loro principale inconveniente è però dato dal fatto che simili istruzioni andrebbero eseguite in formato diretto, cioé da tastiera, prima di aver mandato in esecuzione gli eventuali programmi BASIC che utilizzano la nostra (o le nostre) routine in linguaggio macchina.
In realtà, con un piccolo espediente, è possibile inserire le istruzioni di caricamento dei nostri file o sottoprogrammi internamente a programmi BASIC.
Dovremo, innanzi tutto, essere ben sicuri che il programma BASIC contenente le istruzioni di caricamento non occupi, con parte del suo listato o della sua area dati, la zona in cui il file viene caricato; poiché questo provocherebbe nel primo caso il blocco critico del calcolatore e nel secondo l'interruzione del programma. Nel caso, quindi, in cui il nostro programma sia particolarmente lungo, e sia necessario disporre di una area "tampone" comoda in cui caricare i dati prima del loro definitivo trasferimento nelle aree di memoria loro destinate (ad esempio quelle del terzo banco di memoria, viste nella prima parte dell'articolo), sarà necessario ricorrere ad una tecnica nota in programmazione come "overlay" o "ricopertura" dei programmi.
Questo termine veniva originariamente utilizzato nei sistemi a partizione di memoria o multiutente, in cui più programmi venivano eseguiti, a turno, in parallelo, in modo da suddividere il tempo di elaborazione fra più lavori; dato lo spazio limitato nella memoria centrale (ad accesso veloce) del calcolatore, i lavori in corso di elaborazione, ma non in quel momento in esecuzione, venivano "scaricati" su memoria di massa (come i dischi rigidi) ad accesso più lento rispetto alla memoria centrale, e lo spazio da essi occupato veniva reso disponibile per il programma in esecuzione in quel momento.
Al momento dell'interruzione del programma suddetto gli eventuali programmi "pendenti" potevano, attraverso politiche opportune, venire ricaricati in memoria centrale, "ricoprendo" il programma precedente e riottenendo l'attenzione della unità centrale.
In seguito il termine è passato ad indicare tutti quei casi in cui, non essendoci spazio sufficiente disponibile in memoria, si ricorreva all'artificio di segmentare i programmi, caricandoli quindi nello spazio di memoria disponibile al momento della loro esecuzione.
Nel caso dei piccoli sistemi, come un personal computer del tipo da voi utilizzato, il metodo può rivelarsi particolarmente utile quando vi siano determinati blocchi di istruzioni, che vengono eseguiti una sola volta nel corso del programma (preferibilmente all'inizio) e che interagiscono con il resto del programma stesso solo attraverso il contenuto di determinate locazioni di memoria. Uno degli inconvenienti di questa tecnica è infatti rappresentato dal fatto che i due programmi che si succedono, in "overlay", ad occupare il limitato spazio in memoria disponibile non possono, come suol dirsi, "passarsi" variabili, se non registrandone il contenuto in locazioni di memoria convenute e non interessate dal listato del programma BASIC.
Tutto ciò sembrerà, probabilmente,

piuttosto complicato a chi non abbia mai affrontato prima l'argomento. In pratica non si tratta che di avere un programma BASIC (il programma "caricatore") che si occupi di caricare da nastro o da disco un altro programma BASIC, mandandone quindi in esecuzione la prima istruzione, senza soluzione di continuità, in modo che, virtualmente, sembri di avere a che fare con un unico programma. Il sistema risulta applicabile anche quando si desidera liberare le locazioni iniziali della memoria BASIC (da 2048 in poi), per ospitare dati o routine in linguaggio macchina, e si ritiene noioso digitare ogni volta da tastiera le istruzioni necessarie per spostare i puntatori all'inizio del programma.
La tecnica in questione, infatti, consiste appunto nello specificare il contenuto delle locazioni da 43 a 52, che suddividono la memoria disponibile, in modo da ospitare sia il listato BASIC del programma da caricare sia le variabili, ripartite tra semplici (sia intere che frazionarie), aggregate (matrici) e stringhe.
Le dieci locazioni suddette corrispondono infatti a cinque puntatori, composti da un byte meno significativo ed un byte più significativo. Le locazioni 43 e 44 rappresentano il puntatore all'inizio del programma BASIC (il loro contenuto "per default" è, rispettivamente, 0 e 8; pari a 256*8+0=2048). La coppia 45/46 contiene il puntatore all'inizio delle variabili e, implicitamente, indica anche la fine dell'area di memoria destinata a contenere il listato del programma BASIC, dal momento che le aree suddette sono contigue. Le locazioni 47/48 separano l'area delle variabili semplici dalle variabili aggregate o "array"; il limite superiore dell'area riservata alle matrici (che costituisce, in genere, la parte preponderante della memoria destinata alle variabili) viene segnalata da un puntatore apposito, ospitato dalle locazioni 49 e 50. La fine della memoria viene invece indicata dal puntatore costituito dalle locazioni 51 e 52, che segnalano anche, assieme alla coppia 49/50, l'area disponibile per le variabili alfanumeriche. È interessante notare come, in



questo caso, la registrazione avvenga a partire dagli indirizzi di memoria più alti, "puntati" appunto dalle locazioni in questione, scendendo verso il limite dell'area riservata alle matrici.
Supponendo di dover specificare la seguente ripartizione della memoria

	DA	A
Programma BASIC	833	28159
Variabili semplici	28160	28271
Variabili aggregate	28272	37068
Variabili alfanumeriche	37069	40959

sarà necessario ricorrere alla seguente configurazione di puntatori

Locazione	Contenuto
43	65
44	3
45	0
46	110
47	112
48	110
49	204
50	144
51	255
52	160

Notate come ciascun indirizzo sia stato suddiviso in byte meno significativo (che viene inserito nella locazione del puntatore di indirizzo inferiore) e byte più significativo tramite una semplice divisione con resto, per cui

833:256=3 Resto: 65

Notate, inoltre, come la memoria destinata al programma ed alle sue variabili termini esattamente al di sotto dell'interprete BASIC (da 40960 a 49151). Le eventuali aree disponibili a partire da 49152 possono essere destinate a memoria di schermo o a contenere la forma di eventuali sprite o sottoprogrammi in linguaggio macchina. Come si può notare nell'esempio precedente, infatti, è possibile abbassare il puntatore all'inizio del programma fino a comprendere l'intera memoria di schermo "per default" (da 1024 a 2048) – in questo caso sarà, ovviamente, necessario spostare la memoria di schermo e la memoria carattere nel terzo banco di memoria – e addirittura parte o tutto il buffer del registratore (nel caso in cui non si usi l'unità nastro come periferica) da 828 a 1019.
Nell'esempio in questione le locazioni da 828 a 831 sono state destinate a contenere altrettante variabili (in valore assoluto minori o uguali a 255) che era necessario "passare" al programma "chiamato". È ovviamente possibile aggirare la limitazione riguardante il valore delle variabili, suddividendone il contenuto in più locazioni con la tecnica "byte - più - significativo/byte - meno - significativo".
Per sapere quali valori attribuire ai puntatori è sufficiente interrompere l'esecuzione del programma che si desidera caricare (dopo che siano state dimensionate tutte le variabili aggregate) e far visualizzare al calcolatore il contenuto delle locazioni 43-50, aggiungendo un certo "lasco" per eventuali aggiunte o modifiche al programma.
Naturalmente, è preferibile compiere questa operazione una volta che si sia certi del buon funzionamento del programma chiamato, anche se è ovviamente possibile surdimensionare ogni area in modo opportuno, occupando tutta la memoria disponibile, in modo da lasciare spazio per eventuali aggiunte. Ciò potrebbe addirittura rappresentare un buon sistema di progettazione dei propri programmi: una messa a punto dei singoli blocchi funzionali, seguita da un assemblaggio delle varie parti.
Una volta settati i puntatori suddetti sarà necessario (internamente al programma "chiamante") azzerare il contenuto della locazione a partire da cui si desidera caricare il nuovo programma e della locazione che la precede (nel nostro caso 833 e 832, rispettivamente), quindi bisogna eseguire una istruzione CLR, che azzera il contenuto delle variabili, e infine inserire l'istruzione di caricamento del nuovo programma; ad esempio

```
POKE832,0:POKE833,0: CLR:LOAD
"SECONDA PARTE",8
```

Non è necessario far seguire l'istruzione né da END né da eventuali RETURN o altro. Il calcolatore, infatti, dopo aver caricato il nuovo programma passerà ad eseguire l'istruzione indicata dal puntatore apposito, settato dal programma "caricatore", e proseguirà quindi normalmente.
Nel caso invece in cui si desideri caricare da programma file dati o routine ASSEMBLY direttamente nella zona loro destinata, tramite le istruzioni già viste, non sarà necessario settare puntatori, dal momento che la zona di memoria in cui il programma va caricato viene specificata direttamente all'interno del file stesso.
Il caricamento, però, presenta lo sgradevole inconveniente di azzerare ogni volta il puntatore alla riga BASIC corrente da eseguire. Ciò significa che dopo l'esecuzione di una istruzione del tipo

```
LOAD"PROVA",8,1
```

il programma BASIC che la contiene viene eseguito da capo senza però che le sue variabili vengano azzerate (a meno che contenga una esplicita istruzione di farlo).
Ciò consente di risolvere il nostro piccolo problema. Per comodità e correttezza sarà, innanzi tutto, consigliabile concentrare il caricamento delle routine o dei dati nella parte iniziale del programma, in modo da non lasciare "aperti" eventuali cicli FOR/NEXT o chiamate

a sottoprogrammi.
Supponendo di dover caricare tre file, chiamati "PROVA1", "PROVA2" e "PROVA3", useremo la seguente serie di istruzioni, premettendole a tutte le altre:

```
1IFFL = 0THENFL = 1:LOAD "PROVA1", 8, 1
2IFFL = 1THENFL = 2:LOAD "PROVA2", 8, 1
3IFFL = 2THENFL = 3:LOAD "PROVA3", 8, 1
```

La numerazione del programma effettivo può partire da 4. Vediamo cosa succede al momento di eseguire il nostro programma: dopo averlo mandato in esecuzione con una normale istruzione RUN il contenuto della variabile FL è uguale a zero; il test di riga 1 dà quindi risultato positivo e, dopo aver posto la variabile suddetta uguale a 1, viene caricato il primo file.
Al termine dell'operazione il puntatore alla successiva riga BASIC indica nuovamente la riga 1. Questa volta, però, la variabile FL non è più nulla e quindi si passa ad eseguire la riga 2, dato che il test della 1 è negativo. Sarà ora invece positivo il test contenuto in riga 2 e si provvederà, di conseguenza, ad aggiornare la variabile FL ed a caricare il secondo file. Analogamente si ricicierà alla prima riga, il cui test darà ancora esito negativo, così pure il test di riga 2 (FL è ora uguale a 3) e si passerà quindi ad eseguire la riga 3, il cui test è positivo.
Dopo aver eguagliato a quattro il valore della variabile FL, e caricato il terzo file, il calcolatore si accinge ad eseguire, per la quarta volta, la prima riga del nostro programma. Dato il valore della variabile FL, tutti i test delle prime tre righe risulteranno ora negativi e quindi verrà iniziata l'esecuzione della riga successiva del programma effettivo (con un valore della variabile FL uguale a 4, non a zero).
Il primo programma ASSEMBLY che vedremo consentirà, appunto, di trasferire nelle zone di memoria loro destinate delle routine caricate in un'area tampone da un programma "chiamante", avente questo solo scopo e che si premurerà quindi di caricare il programma effettivo che ne farà uso.

Il trasferimento di blocchi di dati da 256 elementi

Il primo programma ASSEMBLY che vedremo in questa puntata ci consentirà di trasferire blocchi di dati di dimensioni specificate, a patto che esse siano multipli interi di 256. Come abbiamo visto nei numeri scorsi, la parola di memoria del microprocessore 6502, di cui il Commodore 64 dispone, ha una lunghezza di otto bit, che consentono di rappresentare, in formato binario, tutti i numeri decimali compresi tra 0 e 255. Questa limitazione si applica, naturalmente, anche a quei particolari registri del nostro microprocessore che è possibile utilizzare come contatori. È per questa ragione che la maggior parte delle funzioni grafiche, sonore o gestionali del nostro calcolatore è pensata in termini di blocchi di 256 locazioni o multipli e sottomultipli di essi. Così abbiamo visto come gli sprite occupino blocchi di 64 celle di memoria (256/4), l'ultima delle quali ha funzioni semplicemente di "arrotondamento". La stessa memoria di schermo, pur essendo composta da sole mille locazioni (25 righe per 40 colonne), deve iniziare a multipli interi di 1024 (256x4), destinando le ultime sette locazioni del proprio blocco da 1Kbyte ai puntatori alla forma degli sprite. Analogamente la memoria carattere deve occupare blocchi da 2Kbyte (2048 locazioni, 256x8) e deve quindi iniziare in locazioni che siano multipli interi e pari di 2048. Le stesse schermate ad alta risoluzione, che occupano 8Kbyte (8192 byte, 256x32), più le mille locazioni della memoria di schermo, che vengono utilizzate in formato multicolore per specificarne i colori ausiliarii, devono iniziare a multipli pari ed interi di 8192.
L'aspetto interessante del problema è però rappresentato non tanto dagli indirizzi a partire da cui è necessario immagazzinare i dati suddetti, quanto dalle dimensioni dei blocchi di dati stessi, tutte multiple di 256 (nel caso degli sprite, supponendo che essi occupino locazioni di memoria consecutive, è molto più pratico, quando il loro numero sia inferiore a quattro, leggere direttamente i numeri decimali che ne definiscono la forma da frasi DATA, piuttosto che caricarli da nastro).
Ci serviremo di questo particolare per realizzare il nostro programma, che consentirà quindi di trasferire gruppi di dati a blocchi di 256. Prima di affrontare i particolari costruttivi del programma vediamone alcune caratteristiche.
Una prima spiacevole caratteristica è data dalla sua non rilocabilità. Ciò significa, semplicemente, che nel caso in cui desiderassimo porre il nostro programma ASSEMBLY in una parte della memoria diversa da quella in cui lo abbiamo originariamente localizzato non sarebbe sufficiente trasferirne semplicemente, tramite programma assemblatore, il codice nella nuova posizione, ma si renderebbero anche necessarie alcune modifiche ad indirizzi di istruzioni che compongono il programma.
Ciò è dovuto all'uso di istruzioni in formato di indirizzamento assoluto riguardanti locazioni che fanno parte del programma ed il cui indirizzo dipende quindi dalla posizione dello stesso programma.
Una seconda caratteristica è data dall'ordine in cui vengono trasferiti i dati. Contrariamente a quanto visto per la funzione di trasferimento di cui dispone "Supermonitor", il nostro programma effettua la rilocazione delle celle desiderate a partire da quella di indirizzo inferiore. Ciò, naturalmente, non provoca alcun inconveniente nel caso in cui l'area da cui si desidera prelevare i dati e quella in cui si desidera riporli non coincidono nemmeno parzialmente. Nel caso invece in cui alcune locazioni del blocco di partenza siano destinate ad ospitare i dati suddetti anche ad opera-

.A C000 LDY #$00	;azzera il registro Y
.A C002 LDX #$04	;carica nel registro X il numero dei blocchi da 256 elementi che si desidera trasferire (nel nostro caso quattro: 4x256=1024=1Kbyte)
.A C004 LDA $8000,Y	;si carica in accumulatore il contenuto della cella avente indirizzo pari a 8000+Y; 8000 rappresenta la codifica esadecimale dell'indirizzo iniziale del blocco di dati da trasferire
.A C007 STA $6000,Y	;il contenuto dell'accumulatore viene trasferito nella locazione avente indirizzo pari a 6000+Y 6000 rappresenta la codifica esadecimale dell'indirizzo iniziale dell'area destinata ad ospitare i dati che si desidera trasferire
.A C00A INY	;viene incrementato di uno il contenuto del registro Y
.A C00B BNE $C004	;nel caso in cui il flag Z sia diverso da 1 (cioé l'ultima operazione non ha avuto come risultato zero) si ricicla all'istruzione avente l'indirizzo specificato
.A SC00D INC $C006	;viene incrementato il contenuto della cella di indirizzo esadecimale C006 (contenente il byte più significativo dell'indirizzo iniziale del blocco di dati da trasferire)
.A C010 INC $C009	;viene incrementato il contenuto della cella corrispondente al byte più significativo dell'indirizzo di base del nuovo blocco di dati
.A C013 DEX	;viene decrementato di uno il contenuto del registro X
.A C014 BNE $C004	
.A C016 RTS	

Tabella 1

zione conclusa, il trasferimento avverrà senza alcun inconveniente quando ciò avvenga nel verso decrescente degli indirizzi. L'operazione risulterà invece errata per trasferimenti in senso opposto.
Un ultimo particolare: per ragioni di velocità e trasparenza d'uso della routine si è preferito rinunciare alla lettura degli indirizzi di partenza e di arrivo, oltre al numero di blocchi che si desidera trasferire da locazioni convenute. Tali valori vanno quindi inseriti, tramite istruzioni POKE, nelle locazioni del programma destinate ad ospitarli PRIMA dell'esecuzione della routine e, dato che essi vengono modificati nel corso dell'esecuzione stessa, vanno specificati ogni volta, anche nel caso in cui uno o più di essi rimanga invariato tra due esecuzioni successive. Ma vediamo nei particolari il nostro programma (vedi tab. 1), che registreremo a partire dalla locazione 49152 (sarà quindi necessario far uso della versione rilocata di "Supermonitor").

Vediamo di illustrare più dettagliatamente il funzionamento del programma.
La prima istruzione si occupa di azzerare il contenuto del registro Y, che servirà da contatore (internamente ai blocchi di 256 elementi) per il trasferimento dei dati. Perché lo si azzera? Questo registro viene incrementato dopo il trasferimento di ogni cella; questo significa che dopo la prima operazione la configurazione decimale del suo contenuto passa da 0 a 1. Dopo 256 operazioni di trasferimento la sua configurazione binaria passerà da

11111111

a

00000000

il suo contenuto, cioé, sarà nuovamente giunto a zero. Un test su questa configurazione consentirà quindi di stabilire se sia stato trasferito un intero blocco di dati da 256 elementi. Il numero di tali blocchi che si desidera trasferire è specificato dall'istruzione seguente, che carica nel registro X appunto il numero di blocchi da rilocare. Questa istruzione ha una lunghezza di due byte, il primo dei quali (in 49154) ospita il codice operativo della istruzione LDX in formato di indirizzamento immediato, mentre il secondo è destinato a contenere la configurazione corrispondente al numero di blocchi che si desidera trasferire. Questa locazione (49155) rappresenta uno dei valori che devono essere settati prima di ogni esecuzione della nostra routine (anche nel caso in cui si sapesse, a priori, di dover trasferire blocchi di dati sempre della stessa lunghezza, sarebbe necessario ripristinare il contenuto di questa locazione prima di ogni esecuzione, poiché esso viene alterato, come vedremo, nel corso della stessa). Per farlo è sufficiente utilizzare l'istruzione BASIC

POKE49155,NB

dove la variabile NB corrisponderà, di volta in volta, al numero di blocchi voluti. Il fatto che questo valore debba essere contenuto in una parola di memoria da 8 bit ci consente di stabilire l'ampiezza massima dei trasferimenti effettuabili, che sarà, ovviamente, pari a 255 blocchi da 256 elementi, vale a dire 256x255=65280; di gran lunga superiore alle nostre esigenze.
La terza istruzione del nostro programma utilizza il formato di indirizzamento noto come "indicizzato da registro", ottenuto sommando ad un indirizzo di base (contenuto negli ultimi due byte dei tre che l'istruzione occupa) il contenuto di un registro che viene specificato dal codice operativo contenuto nel primo byte dell'istruzione. Nel nostro caso il registro in questione è il registro Y, mentre l'indirizzo di base ha codifica (esadecimale) $8000, pari a 32768 decimale. Il contenuto della locazione corrispondente all'indirizzo di base, cui viene sommato il valore attuale del registro Y, viene caricato in accumulatore. Le due locazioni contenenti l'indirizzo di base (nel nostro caso 49157 e 49158) andranno settate prima di ogni operazione di trasferimento, in base all'indirizzo iniziale del blocco di dati che si desidera trasferire. Tale indirizzo andrà suddiviso in byte più significativo (pari a indirizzo/256), che andrà inserito in 49158, e byte meno significativo (il resto della operazione precedente), da inserire in 49157. Le due istruzioni necessarie per questo scopo sono le seguenti (supponendo di aver indicato con la variabile IB l'indirizzo di base):

POKE49158, INT(IB/256)

POKE49157, IB-INT(IB/256)*256

L'istruzione successiva trasferisce il contenuto dell'accumulatore nella locazione avente indirizzo pari a quello iniziale della nuova area destinata ad ospitare i dati, cui viene sommato il contenuto del registro Y.
Il formato è analogo a quello dell'istruzione precedente: tre byte con indirizzamento indicizzato da registro; il primo byte contiene il codice operativo dell'istruzione (che specifica il verso del trasferimento ed il registro coinvolto), mentre i byte seguenti contengono l'indirizzo suddetto nel solito formato "byte - meno - significativo/byte - più - significativo". Nel nostro caso il byte meno significativo andrà, di volta in volta, inserito nella locazione di indirizzo decimale 49160, mentre il più significativo verrà ospitato dalla locazione seguente. Per questa istruzione valgono tutte le considerazioni fatte per la precedente. Notate come gli indirizzi delle locazioni suddette siano caratteristici della forma attuale assunta dal nostro programma; nel caso, quindi, in cui si desideri rilocarne il codice sarà necessario modificare di conseguenza anche gli indirizzi in cui caricare gli indirizzi di base dell'area sorgente e della destinazione. Nel caso in cui si trasferisca il programma suddetto da 49152 a 955 sarà necessario sottrarre ad ogni indirizzo lo scarto tra i due indirizzi. Pertanto la locazione 49155, che, come abbiamo visto, contiene il numero di blocchi che si desidera trasferire, diverrà la 49155 − (49152−955) = 958 e così via per tutte le altre locazioni.
Dopo le due istruzioni precedenti il contenuto del registro Y viene incrementato e viene quindi eseguito un test per stabilire se l'istruzione in questione abbia provocato l'attivazione del bit Z del registro dei flag, che segnala appunto se una operazione ha risultato nullo. In caso negativo il trasferimento del blocco da 256 dati non è stato portato a termine, pertanto si ricicla alla istruzione di caricamento in accumulatore del dato da trasferire. Il formato di questa istruzione, pur specificando l'indirizzo dell'istruzione a cui si desidera riciclare se il test ha esito negativo, utilizza un ulteriore formato di indirizzamento, chiamato "relativo", che viene adottato ogni volta in cui la destinazione di un salto logico si trovi all'interno di un blocco da 256 caratteri, centrato sulla locazione successiva a quella contenente l'istruzione di test stessa. Infatti, poiché lo spostamento può essere positivo o negativo (cioé un salto verso una istruzione successiva o precedente quella attuale), i valori numerici compresi tra 0 e 128 indicano una diramazione diretta, mentre quelli compresi tra 129 e 255 una diramazione inversa, la cui ampiezza è data dal complemento a 255 del valore specificato. Nel nostro caso lo scarto tra la locazione contenente il secondo byte dell'istruzione BNE $C004 e la destinazione della diramazione stessa è pari a 49164−49156=8. Trattandosi di un salto "all'indietro", cioè di una diramazione inversa, il valore da



specificare sarà pari a 255–8=247 ($E7).
Non è necessario, comunque, che vi preoccupiate di calcolare le entità delle diramazioni: il nostro programma assemblatore provvederà a farlo per voi; sarà sufficiente indicare l'indirizzo della destinazione.
Nel caso in cui il test dell'istruzione precedente abbia avuto esito positivo, cioè un intero blocco da 256 dati sia stato trasferito, sarà necessario aggiornare il valore degli indirizzi di base, sia dell'area dati da cui i dati vengono trasferiti che della loro destinazione. Notate come il contenuto del registro-contatore Y sia già posto al valore opportuno (cioè zero).
Questa operazione di incremento del byte più significativo dei due indirizzi suddetti rappresenta un altro punto critico del nostro programma.
Questi indirizzi sono infatti, come abbiamo visto, caratteristici del programma. I valori da inserire nel codice ASSEMBLY della nostra routine dipenderanno quindi dalla posizione in memoria che essa occupa. In caso di rilocazione valgono di conseguenza tutte le considerazioni fatte in precedenza.
Queste due istruzioni (entrambe a tre byte; gli indirizzi andranno perciò suddivisi nel modo consueto) fanno sì che il programma non sia immediatamente rilocabile tramite una semplice operazione di trasferimento del codice ASSEMBLY che lo compone. Gli indirizzi contenuti in esse dovranno sempre essere "corretti", dopo aver trasferito il programma nella nuova posizione, in modo da puntare all'ultimo byte delle istruzioni

```
LDA $Indirizzo iniziale,Y
```

e

```
STA $Indirizzo finale,Y
```

Dopo aver aggiornato i due indirizzi di base, in modo che puntino al successivo blocco da 256 locazioni, è necessario decrementare il contenuto del registro X, che, ricorderete, ospitava il numero dei blocchi da 256 che si voleva trasferire. Viene quindi eseguito un nuovo test sul contenuto del bit Z del registro di flag. Nel caso, infatti, in cui l'operazione precedente abbia azzerato il contenuto del registro X l'operazione di trasferimento si può considerare conclusa: il numero prefissato di blocchi da 256 locazioni che si desiderava rilocare è stato trasferito. Proseguendo in sequenza si incontra la consueta istruzione RTS (ReTurn from Subroutine, rientro da sottoprogramma), che consente di tornare in ambiente BASIC.
In caso contrario si ricicla alla istruzione di lettura della nuova locazione da trasferire con il contatore Y azzerato ed il contenuto del registro X indicante il numero di blocchi rimanenti da trasferire. Anche in questo caso, come in quello già visto, il formato dell'istruzione è relativo. L'entità del salto, in questo caso, è pari a 49173–49156=17 e, trattandosi di diramazione inversa, il valore indicato nel secondo byte dell'istruzione sarà, in formato decimale, 255–17=238 ($EE).
Supponendo quindi di dover trasferire un blocco di dati di lunghezza pari a 2Kbyte (2048 elementi) da 32768 in 57344, le istruzioni necessarie sarebbero le seguenti

```
POKE49155, 2048/256
POKE49157, 0:POKE49158, 128
POKE49160, 0:POKE49161, 224
SYS49152
```

L'esempio precedente potrebbe riguardare il trasferimento dei dati necessari a ridefinire il set di caratteri di cui il calcolatore dispone, caricati da nastro o da disco tramite il metodo rapido visto nel primo paragrafo e registrati in origine a partire dalla locazione 32768.
Ovviamente, nel caso in cui il programma destinato ad ospitare i dati abbia dimensioni tali da interferire con l'area "tampone" indicata, le operazioni di caricamento e trasferimento dovrebbero venire svolte da un programma "caricatore", che si premurerebbe, quindi, di richiamare, in overlay, il programma utente.

Una routine per il trasferimento di blocchi di lunghezza prefissata e diversa da 256 locazioni

Il programma ASSEMBLY presentato nel paragrafo precedente consentiva di trasferire blocchi di dati di lunghezza voluta, a patto che questa fosse multipla di 256. Può accadere, per svariate ragioni, che l'ordine del trasferimento dei dati, pur essendo sequenziale, non possa prescindere dalle caratteristiche intrinseche dei dati che si desidera trasferire. Ad esempio, dovendo trasferire il contenuto dell'area dati che rappresenta la memoria di schermo o la memoria colore, in modo da effettuarne lo scrolling in una determinata direzione, non è possibile utilizzare blocchi da 256 dati, ma sarà necessario servirsi di blocchi di dimensioni inferiori.
La gestione della memoria grafica del Commodore 64 presenta una ulteriore particolarità: non ci si deve occupare solo di operare sul contenuto della memoria di schermo, come per i vecchi calcolatori della linea Commodore (i Pet 2001, i CBM 3032 e 4032), per i quali, essendo monocromatici, l'argomento grafica si esauriva con la conoscenza dell'indirizzo di base della memoria destinata allo schermo (32768, detto per inciso). I nuovi "home computer" della Commodore, i VIC 20, i C64 e C64 plus, il C16, dispongono di caratteristiche grafiche più evolute e, oltre a poter spostare la memoria destinata allo schermo internamente allo spazio di memoria disponibile, prevedono anche la possibilità di gestire formati di grafica multicolore. Ciò significa, ovviamente, che dovremo occuparci anche del trasferimento della memoria colore del nostro calcolatore, se non vorremo veder le nostre schermate cambiare colore nel bel mezzo dello scrolling.

.A C000 LDA #$27	;viene caricato in accumulatore il valore corrispondente al byte meno significativo dell'indirizzo di base iniziale per la memoria di schermo
.A C002 STA $4E	;il contenuto dell'accumulatore viene registrato nella locazione 78 di pagina 0.
.A C004 LDA #$04	;byte più significativo dell'indirizzo suddetto
.A C006 STA $4F	;ACCUM → 79
.A C008 LDA #$27	;byte meno significativo dell'indirizzo di base iniziale della memoria colore
.A C00A STA $52	;ACCUM → 82
.A C00C LDA #$D8	;byte più significativo dell'indirizzo precedente
.AC00E STA $53	;ACCUM → 83
.A C010 LDA #$FF	;byte meno significativo dell'indirizzo di base della destinazione della memoria di schermo
.A C012 STA $50	;ACCUM → 80
.A C014 LDA #$03	;byte più significativo dello stesso indirizzo
.A C016 STA $51	;ACCUM → 81
.A C018 LDA #$FF	;byte meno significativo dell'indirizzo di base della destinazione della memoria colore
.A C01A STA $57	;ACCUM → 87
.A C01C LDA #$D7	;byte più significativo dell'indirizzo in questione
.A C01E STA $58	;ACCUM → 88
.A C020 LDX #$18	;viene caricato nel registo X il numero delle righe che si desidera trasferire ($18 = #24)
.A C022 LDY #$28	;nel registro Y si carica il numero di caratteri che compongono ciascuna riga
.A C024 LDA ($4E),Y	viene caricato in accumulatore il contenuto della cella il cui indirizzo è registrato nella coppia di byte 78/79 (in cui 78 rappresenta il byte meno significativo) a cui viene sommato il contenuto attuale del registro Y

Tabella 2 **(segue)**

```
.A C026 STA ($50),Y    ;il contenuto dell'accumulatore viene trasferito
                       nella locazione avente indirizzo corrispondente al
                       contenuto della coppia di locazioni 80/81, cui vie-
                       ne sommato il registro contatore Y
.A C028 LDA ($52),Y    ;il contenuto della locazione avente indirizzo pari
                       alla coppia di byte 82/83, più il registro Y, viene
                       caricato in accumulatore
.A C02A STA ($57),Y    ;il contenuto dell'accumulatore viene trasferito
                       nella locazione il cui indirizzo è registrato nella
                       coppia 87/88 ed a cui viene sommato il valore
                       attuale del registro Y
.A C02C DEY            ;il contenuto del registro Y viene decrementato di
                       una unità
.A C02D BNE $C024      ;se il contenuto del registro Y è diverso da zero
                       (cioè il bit Z del registro di flag, dopo l'operazione
                       precedente, è nullo) si ricicla all'istruzione corri-
                       spondente all'indirizzo specificato
.A C02F CLC            ;viene azzerato il bit di riporto del registro di flag
.A C030 LDA $4E        ;viene trasferito in accumulatore il contenuto at-
                       tuale della locazione 78
.A C032 ADC #$28       ;viene effettuata la somma con riporto del conte-
                       nuto dell'accumulatore più il valore numerico indi-
                       cato ($28 = #40), spostando l'indirizzo di base
                       di una riga verso il basso (nel caso di scrolling
                       dall'alto in basso sarà necessario, oltre che cam-
                       biare gli indirizzi di base iniziale e finale per en-
                       trambe le memorie, sottrarre tale valore numerico
                       invece di sommarlo)
.A C034 STA $4E        ;il contenuto dell'accumulatore viene reimmagaz-
                       zinato nella locazione 78
.A C036 LDA $00        ;si azzera l'accumulatore
.A C038 ADC $4F        ;all'accumulatore si aggiunge il contenuto della
                       locazione 79, cioè il byte più significativo dell'indi-
                       rizzo di base iniziale della memoria di schermo
.A C03A STA $4F        ;il risultato dell'operazione precedente (compreso
                       il riporto della somma precedente, che non è stato
                       azzerato) viene registrato nella locazione 79. Le
                       sei istruzioni precedenti, insieme a questa stessa,
                       rappresentano un esempio di somma a sedici bit
.A C03C CLC            ;Carry ← "0"
.A C03D LDA $50        ;ACCUM ← [80]
.A C03F ADC #$28       ;ACCUM+40 → ACCUM
.A C041 STA $50        ;ACCUM → [80]
.A C043 LDA #$00       ;ACCUM ← "0"
.A C045 ADC $51        ;ACCUM+[81]+Carry → ACCUM
.A C047 STA $51        ;ACCUM → [81]
.A C049 CLC            ;Carry ← "0"
.A C04A LDA $52        ;ACCUM ← [82]
.A C04C ADC #$28       ;ACCUM+40 → ACCUM
.A C04E STA $52        ;ACCUM → [82]
.A C050 LDA #$00       ;ACCUM ← "0"
.A C052 ADC $53        ;ACCUM+[83]+Carry → ACCUM
.A C054 STA $53        ;ACCUM → [83]
.A C056 CLC            ;Carry ← "0"
.A C057 LDA $57        ;ACCUM ← [87]
.A C059 ADC #$28       ;ACCUM+40 → ACCUM
.A C05B STA $57        ;ACCUM → [87]
.A C05D LDA #$00       ;ACCUM ← "0"
.A C05F ADC $58        ;ACCUM+[88]+Carry → ACCUM
.A C061 STA $58        ;ACCUM → [88]
.A C063 DEX            ;il contenuto del registro X viene decrementato di
                       una unità
.A C064 BNE $C022      ;nel caso in cui il contenuto del registro X sia diver-
                       so da zero (bit Z del registro di flag disattivato), si
                       ricicla a partire dalla istruzione di inizializzazione
                       del registro Y a 40
.A C066 RTS            ;si rientra in ambiente BASIC
```

L'esempio che mostreremo in questo paragrafo opererà lo scrolling nella direzione consueta (dal basso verso l'alto), una riga di schermo ad ogni esecuzione della nostra subroutine ASSEMBLY. Le modifiche da apportare al programma per ottenere lo scrolling nella direzione opposta consistono, semplicemente, in una diversa scelta dell'indirizzo di base iniziale e dell'indirizzo di base finale, oltre che in un ciclo che provveda a decrementare gli indirizzi invece di incrementarli, come nel programma seguente

Indirizzamento indiretto, indicizzato indiretto, indiretto indicizzato e... spie

Commentiamo diffusamente anche il listato di questo secondo programma ASSEMBLY, il che ci permetterà di introdurre anche nuovi formati di indirizzamento.

Le prime sedici istruzioni del programma sono destinate all'inizializzazione delle otto locazioni di pagina zero destinate a contenere (suddivisi in byte più e meno significativo) gli indirizzi di base iniziali e finali per le due zone di memoria interessate dal trasferimento: la memoria di schermo e la memoria colore.

Le prime quattro istruzioni trasferiscono il byte meno significativo (39) dell'indirizzo di partenza della memoria di schermo in 78 ed il byte più significativo (4) in 79. L'indirizzo di base della memoria di schermo è quindi dato da 4*256+39=1063. Dato che i registri contatori da noi utilizzati puntano esattamente al di sotto del blocco da trasferire, ciò farà sì che si cominci col trasferire la seconda riga della memoria di schermo nella sua configurazione attuale.

Le successive quattro istruzioni trasferiscono l'indirizzo di base iniziale della memoria colore nella coppia di locazioni di pagina zero 82/83 (il byte meno significativo verrà ospitato da 82). Notate come ciascun indirizzo venga registrato in una coppia di byte consecutivi; ciò ci permetterà di considerare le singole coppie di locazioni come altrettanti indirizzi di locazioni prefissate, leggibili con un formato di indirizzamento che non abbiamo ancora introdotto: l'indirizzamento indiretto.

Ciò equivale ad avere l'indirizzo di una determinata locazione posto in una posizione di memoria convenuta. Per usare un altro termine, mutuato dalla gestione dei sistemi operativi, ciò significa utilizzare una tecnica a "mailbox", cioè a "caselle postali", in cui determinate locazioni svolgono il ruolo di tramite di informazioni tra diverse routine o tra parti di una stessa routine. Analogamente agli agenti dei racconti di spionaggio che lasciano i microfilm in un nascondiglio convenuto perché un corriere passi poi a ritirarli, o ad una famiglia in cui ciascun componente sa dove guardare nel caso in cui sia sprovvisto di chiavi e non ci sia nessuno in casa (tipicamente, la chiave è sotto lo stuoino o dietro il vaso); così nel nostro programma gli indirizzi di cui la routine deve servirsi sono in locazioni prefissate e di facile accesso (in pagina zero, la qual



cosa, come abbiamo già visto, consente una maggior velocità di elaborazione; determinante nel caso in cui i dati da trattare siano molti ed abbiano direttamente a che fare con la gestione della grafica del calcolatore).

Le locazioni di pagina zero utilizzate devono essere, come detto, consecutive. Ciò ci permetterebbe di disporre di due soli indirizzi, nelle locazioni di pagina zero inutilizzate, citate nel numero scorso: 251/252 e 253/254.

Il Commodore 64, però, dispone in pagina zero di due aree "scratch", il cui contenuto, cioè, è variabile e, per così dire, "volatile". Non ha quindi importanza se vengono apportati cambiamenti al loro contenuto, poiché il loro significato dipende dalla routine di sistema operativo che ne fa uso, sia come memoria "tampone" sia come area di lavoro temporanea, per ospitare operandi o dati provenienti da puntatori ed in corso di trasferimento. Le due aree si trovano in 78-83 ed in 87-96.

La prima delle due è destinata agli usi più disparati, mentre la seconda è riservata come area di lavoro per operazioni numeriche. Entrambe possono essere comunque utilizzate, a patto che il loro contenuto venga usato direttamente nel corso della stessa subroutine, poiché, nel caso in cui fra una routine che impiega le locazioni suddette e la successiva vengano utilizzate istruzioni BASIC che comportano l'uso delle aree in questione, il loro contenuto verrebbe, inevitabilmente, alterato; non è possibile ripiegare sulle meno veloci, ma più affidabili, locazioni al di fuori della pagina zero, poiché il microprocessore 6502 non ha una capacità di indirizzamento indiretto completamente generale, ma limita il campo indirizzo agli otto bit necessari per accedere alla pagina zero.

Le ultime otto istruzioni di questo primo blocco inizializzano gli indirizzi di base finali per le due memorie coinvolte nell'operazione.

L'indirizzo di base della memoria di schermo viene inizializzato a 3*256+255=1023 (esattamente al di sotto della memoria di schermo visibile) e posto in 80/81, e la memoria colore a 215*256+255=55295 (40 byte, cioè un'intera riga al di sotto dell'indirizzo di base iniziale 216*256+39=55335), in 87/88.

Le due istruzioni successive inizializzano i registri contatori X e Y, destinati a contenere, rispettivamente, il numero di blocchi (righe) da trasferire e la lunghezza di ciascun blocco.

Come potete notare dal listato, mentre il registro Y è stato inizializzato alla effettiva lunghezza delle righe di schermo, il numero di righe non corrisponde al totale di quelle visualizzate (25). La tecnica dello scrolling consiste appunto, in questo caso, nel sovrapporre il contenuto delle ultime ventiquattro righe sulle prime ventiquattro.

Le quattro istruzioni che seguono utilizzano l'indirizzamento indiretto nella sua forma indicizzata dal registro. È necessaria una piccola precisazione, per evitare l'insorgere di un banale equivoco, molto diffuso in campo programmativo.

È necessario distinguere tra indirizzamento indicizzato indiretto ed indiretto indicizzato. Il primo è quello da noi qui utilizzato e consiste nel reperire l'indirizzo di base di una determinata area dati in una coppia di locazioni di pagina zero (indirizzamento indiretto) e sommarvi un indice ricavabile da un registro (indirizzamento indicizzato).

Il secondo formato consiste nel sommare il valore dell'indice ad un indirizzo di base che fa riferimento all'area di pagina zero della "casella postale", contenente gli indirizzi effettivi desiderati; ciò equivale ad una indicizzazione seguita da una indirezione.

Le quattro operazioni in questione consistono nel leggere il contenuto della casella il cui indirizzo è ottenuto sommando ad un indirizzo di base, specificato nella coppia di locazioni 78/79 in pagina zero (per indicare le quali è sufficiente specificare l'indirizzo del byte meno significativo), il contenuto del registro contatore Y e nel trasferirlo in quella il cui indirizzo è ricavato con analogo meccanismo dall'indirizzo di base finale registrato in 80/81. Le seconde due istruzioni svolgono compiti analoghi per le locazioni della memoria colore.

Al termine delle operazioni precedenti viene decrementato il registro contatore Y e si esegue un test tendente a stabilire se sia stata trasferita un'intera riga. In caso negativo si ricicla a partire dalla prima locazione in indirizzamento indicizzato indiretto (o, come dicono alcuni testi, post-indicizzato).

Se è stata invece trasferita un'intera riga, si aggiornano i contenuti delle locazioni di pagina zero che ospitano gli indirizzi, ricordando che si tratta di indirizzi a 16 bit, composti cioé da due byte, con tutto quel che ciò comporta per quanto riguarda la gestione dei riporti tra il primo e il secondo byte.

Le ventotto istruzioni che seguono il test si possono infatti suddividere in quattro gruppi, ognuno dei quali aumenta di quaranta (la lunghezza di una riga di schermo) ciascuno degli indirizzi di base utilizzati.

Sinteticamente, ciascun gruppo è composto dall'azzeramento del bit di riporto (Carry) del registro di flag, dalla lettura del byte meno significativo dell'indirizzo (contenuto nel primo dei due byte di ogni coppia), da una istruzione ADC (Add with Carry, somma con riporto), tra il contenuto dell'accumulatore ed il byte meno significativo della cifra esadecimale 40; il contenuto dell'accumulatore viene quindi ritrasferito nella locazione di pagina zero destinata a contenere il byte meno significativo di ciascun indirizzo (eventuali riporti sono contenuti nel bit di Carry).

La seconda serie di operazioni del blocco di aggiornamento consiste nel sommare al byte più significativo di ciascun indirizzo il byte più significativo della cifra esadecimale suddetta, cioè zero. Questa operazione permette di incorporare nella somma eventuali riporti tra il primo ed il secondo byte che compongono l'indirizzo. Per farlo viene azzerato l'accumulatore e ad esso viene sommato (sempre con riporto) il contenuto della locazione voluta (indirizzamento "in pagina zero"). Il risultato viene quindi trasferito nella locazione destinata ad ospitare il byte più significativo dell'indirizzo in questione.

La serie di operazioni suddette non può venire svolta tramite un ciclo, sia per la mancanza di un registro da utilizzare come contatore (X ed Y sono già impegnati), sia per la non consecutività delle locazioni che compongono la "mailbox" (la coppia 87/87 è separata dalle altre dall'area di pagina zero destinata a contenere i vettori di salto logico per le funzioni).

Le penultime due istruzioni del programma decrementano il registro X (numero delle righe da trasferire) ed eseguono il consueto test, per stabilire se l'operazione è da considerarsi completata per tutte le righe volute. In caso negativo si ricicla alla riga che inizializza a 40 il registro Y e si prosegue, come già visto. In caso contrario si rientra in ambiente BASIC tramite la consueta istruzione RTS.



MLX per VIC 20 e C64 (versione 2.0 per C64)

di C. Brannon
trad. e adatt.
di M. Cristuib Grizzi e F. Stella

MLX è un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e quello della locazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato. MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali:

SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed è utile nel caso si inseriscano i listati in più parti.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o

SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file",1,1 per il registratore, oppure LOAD "nome del file",8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

La versione originaria di MLX per C64 ha subito, dalla prima pubblicazione, diversi ritocchi e miglioramenti, fino alla versione 2.0 qui presentata.

È stato in primo luogo modificato il colore di fondo dello schermo e del bordo, in modo da risultare meno stancante alla vista; sono state ulteriormente perfezionate le routine in linguaggio macchina di salvataggio e caricamento dei programmi; è cambiata la forma del cursore e, cosa più importante, è stato aggiunto un tastierino numerico per gli utenti che si trovano più a loro agio con una diversa disposizione dei tasti.

Oltre a poter usare i soliti tasti numerici, sono stati ridefiniti alcuni tasti alfabetici, in modo da generare ugualmente dei numeri. I tasti ridefiniti sono i seguenti:

U I O		7 8 9
H J K L	diventano	0 4 5 6
M , .		1 2 3

Le persone abituate ad usare tali tastierini accresceranno senz'altro, in brevissimo tempo, la velocità di battitura dei programmi in linguaggio macchina.

MLX versione per VIC 20

```
100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
                               :rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
                               :rem 144
110 PRINT"{RVS}{ 14 SPAZI}"    :rem 117
120 PRINT"{RVS} {DES}{OFF}[<*>]£{RVS}
    {DES} {DES}{ 2 SPAZI}[<*>]{OFF}[<*>]
    £{RVS}£{RVS} "             :rem 191
130 PRINT"{RVS} {DES} [<G>]{DES}
    { 2 DES} {OFF}£{RVS}£[<*>]{OFF}[<*>]
    {RVS} {OFF}"               :rem 122
140 PRINT"{RVS}{ 14 SPAZI}"    :rem 120
200 PRINT"{ 2 GIU'}{PUR}{BLK}UN PROGRAMMA
     PER L'IN-TRODUZIONE{ 2 SPAZI}DI ROUT
    INEIN LINGUAGGIO MACCHINA"; :rem 202
205 PRINT"A PROVA DI ERRORE{ 5 GIU'}"
                               :rem 226
210 PRINT"{BLK}{ 4 SU}INDIRIZZO DI PARTEN
    ZA":INPUTS:F=1-F:C$=CHR$(31+119*F)
                               :rem 52
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
                               :rem 2
225 PRINT:PRINT:PRINT:PRINT    :rem 123
230 PRINT"{BLK}{ 3 SU}INDIRIZZO CONCLUSIV
    O":INPUTE:F=1-F:C$=CHR$(31+119*F)
                               :rem 170
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
                               :rem 234
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
    LUSIVO <INDIRIZZO DI PARTENZA
    { 3 GIU'}":GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT          :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S :rem 56
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6           :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                               :rem 228
390 IFN=-211THEN710            :rem 62
400 IFN=-204THEN790            :rem 64
410 IFN=-206THENPRINT:PRINT"{GIU'}INSERIS
    CI UN NUOVO{ 4 SPAZI}INDIRIZZO";:INPU
    TZZ                        :rem 42
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}ESCE DAL CAMPO DI VA-" :rem 150
416 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}LORI INDICATO":GOSUB1000:GOTO410
                               :rem 105
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                               :rem 238
420 IFN<>-196THEN480           :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT,"
    { 2 SIN}A";:INPUTT         :rem 29
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
    ;S;"{SIN}, MASSIMO";E:GOTO430
                               :rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":"; :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
    PC(10);                    :rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
    ,";                        :rem 157
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                               :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                  :rem 50
480 IFN<0THENPRINT:GOTO310     :rem 168
490 A(J)=N:NEXTJ               :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                               :rem 234
515 IFN=CKSUMTHEN530           :rem 255
520 PRINT:PRINT"LA RIGA E' STATA INSE-RIT
    A IN MANIERA ERRATA"       :rem 39
525 PRINT"RI-INSERIRLA":PRINT:GOSUB1000:G
    OTO310                     :rem 167
530 GOSUB2000                  :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
                               :rem 80
550 AD=AD+6:IFAD<ETHEN310      :rem 212
560 GOTO710                    :rem 108
570 N=0:Z=0                    :rem 88
```

```
580 PRINT"[<+>]";                        :rem  79
581 GETA$:IFA$=""THEN581                 :rem  95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                        :rem 229
590 IFA>128THENN=-A:RETURN               :rem 137
600 IFA<>20THEN630                       :rem  10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {SIN} {SIN}";:GOTO690                :rem 172
620 GOTO570                              :rem 109
630 IFA<48ORA>57THEN580                  :rem 105
640 PRINTA$;:N=N*10+A-48                 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
                                         :rem 229
660 Z=Z+1:IFZ<3THEN580                   :rem  71
670 IFZ=0THENGOSUB1000:GOTO570           :rem 114
680 PRINT",";:RETURN                     :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                         :rem 149
692 FORI=1TO3:T=PEEK(S%-I)               :rem  68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                         :rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
                                         :rem   7
710 PRINT"{CLR}{RVS}*** SAVE ***
    { 3 GIU'}"                           :rem 236
720 INPUT"{GIU'} NOME DEL FILE";F$
                                         :rem 199
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"            :rem 128
740 GETA$:IFA$<>"N"ANDA$<>"D"THEN740
                                         :rem  30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                         :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                     :rem   3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                         :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                   :rem  69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                         :rem  12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                            :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                         :rem 111
775 PRINT"{GIU'}OPERAZIONE CONCLUSA.":END
                                         :rem 152
780 PRINT"{GIU'}ERRORE DI REGISTRAZIO-NE.
    RIPROVA.":IFDV=1THEN720              :rem   8
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720                 :rem 103
782 GOTO720                              :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***
    { 2 GIU'}"                           :rem 212
800 INPUT"{ 2 GIU'} NOME DEL FILE";F$
                                         :rem 215
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"            :rem 127
820 GETA$:IFA$<>"N"ANDA$<>"D"THEN820
                                         :rem  28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                         :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                     :rem   2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                         :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                   :rem  70
850 POKE780,0:SYS65493                   :rem  11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                         :rem 111
865 PRINT"{GIU'}OPERAZIONE CONCLUSA.":GOT
    O310                                 :rem 142
870 PRINT"{GIU'}ERRORE DI CARICAMENTO.RIP
    ROVA.{GIU'}":IFDV=1THEN800           :rem  43
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800                 :rem 102
1000 REM CICALINO                        :rem 231
1001 POKE36878,15:POKE36874,190          :rem 206
1002 FORW=1TO300:NEXTW                   :rem 117
1003 POKE36878,0:POKE36874,0:RETURN
                                         :rem  74
2000 REM CAMPANELLO                      :rem 130
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
     76,240:NEXTW                        :rem  22
2002 POKE36876,0:RETURN                  :rem 119
3000 PRINTC$;"{RVS}NON IN PAGINA ZERO O
     { 2 SPAZI}SU ROM{ 4 GIU'}":GOTO1000
                                         :rem  23
```

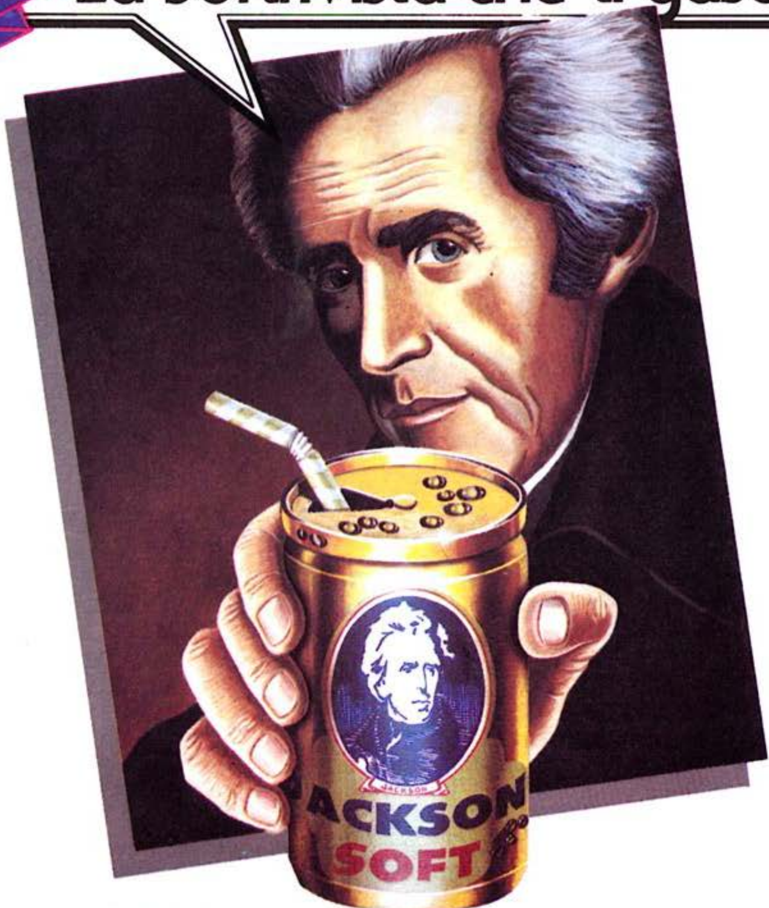
MLX versione 2.0 per C64

```
100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
    POKE53280,0:POKE53281,0              :rem  71
101 POKE788,52:REM DISABILITA RUN/STOP
                                         :rem  89
110 PRINT"{RVS}{ 40 SPAZI}";             :rem 176
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}[<*>]
    ~{RVS}{DES} {DES}{ 2 SPAZI}[<*>]{OFF}
    [<*>]~{RVS}~{RVS}{ 13 SPAZI}";
                                         :rem 250
130 PRINT"{RVS}{ 15 SPAZI}{DES}[<N>][<H>]
    {DES} { 2 DES} {OFF}~{RVS}~[<*>]{OFF}
    [<*>]{RVS}{ 13 SPAZI}";              :rem 220
140 PRINT"{RVS}{ 40 SPAZI}"              :rem 120
200 PRINT"{ 2 GIU'}{PUR}EDITOR DI CODICE
    MACCHINA VERSIONE 2.0{ 3 GIU'}"
                                         :rem 126
210 PRINT"[<5>]{ 2 SU} INDIRIZZO DI PARTE
    NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
    31+119*F)                            :rem  85
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210                :rem 235
225 PRINT:PRINT:PRINT                    :rem 180
230 PRINT"[<5>]{ 2 SU} INDIRIZZO CONCLUSI
    VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$(3
    1+119*F)                             :rem  92
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230                :rem 183
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
    LUSIVO<INDIRIZZO INIZIALE"           :rem 200
255 IFE<STHENGOSUB1000:GOTO230           :rem 119
260 PRINT:PRINT:PRINT                    :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                         :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";:FORJ=ATO6                 :rem 227
315 FORJ=ATO6                            :rem  33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                         :rem 228
390 IFN=-211THEN710                      :rem  62
400 IFN=-204THEN790                      :rem  64
410 IFN=-206THENPRINT:INPUT"{GIU'} INSERI
    RE IL NUOVO INDIRIZZO";ZZ            :rem  13
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}ESCE DAL CAMPO DI VALORI INDICAT
    O"                                   :rem 105
```

```
415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000
    :GOTO410                      :rem  202
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                  :rem  238
420 IFN<>-196THEN480              :rem  133
430 PRINT:INPUT"LISTATO:DA";F:PRINT"
    { 9 SPAZI}A";:INPUTT          :rem  183
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
    ;S;" MASSIMO";E;"![<5>]":GOTO430
                                  :rem  147
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";    :rem   30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";     :rem   66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                  :rem   25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                     :rem   50
480 IFN<0THENPRINT:GOTO310        :rem  168
490 A(J)=N:NEXTJ                  :rem  199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT :rem  200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                            :rem   94
511 IFN=-1THENA=6:GOTO315         :rem  254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                  :rem  122
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN
    SERITA IN MANIERA"            :rem  157
525 PRINT"ERRATA. RIPETERE[<5>]":PRINT:GO
    SUB1000:GOTO310               :rem   27
530 GOSUB2000                     :rem  218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0             :rem  227
550 AD=AD+6:IFAD<ETHEN310         :rem  212
560 GOTO710                       :rem  108
570 N=0:Z=0                       :rem   88
580 PRINT"[<~>]";                 :rem   81
581 GETA$:IFA$=""THEN581          :rem   95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L")
                                  :rem   41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="0"          :rem  134
584 IFAV>0THENA$=CHR$(48+AV)      :rem  134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                 :rem  229
590 IFA>128THENN=-A:RETURN        :rem  137
600 IFA<>20THEN630                :rem   10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {SIN} {SIN}";:GOTO690         :rem  172
620 GOTO570                       :rem  109
630 IFA<48ORA>57THEN580           :rem  105
640 PRINTA$;:N=N*10+A-48          :rem  106
650 IFN>255THENA=20:GOSUB1000:GOTO600
                                  :rem  229
660 Z=Z+1:IFZ<3THEN580            :rem   71
670 IFZ=0THENGOSUB1000:GOTO570    :rem  114
680 PRINT",";:RETURN              :rem  240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                  :rem  149
691 FORI=1TO3:T=PEEK(S%-I)        :rem   67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                  :rem  205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
                                  :rem    7
710 PRINT"{CLR}{RVS}*** SAVE ***
    { 3 GIU'}"                    :rem  236
715 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
    {OFF} PER USCIRE DAL SAVE{GIU'}"
                                  :rem  103
720 F$="":INPUT"{GIU'}NOME DEL FILE";F$:I
    FF$=""THENPRINT:PRINT:GOTO310
                                  :rem   42
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"    :rem  128
740 GETA$:IFA$<>"N"ANDA$<>"D"THEN740
                                  :rem   30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15    :rem  212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem    3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                  :rem  109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem   69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE780,253              :rem   17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496               :rem  235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                  :rem  111
775 PRINT"{GIU'}FATTO.{GIU'}":GOTO310
                                  :rem  201
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!"
    :IFDV=1THEN720                :rem  104
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720          :rem  103
790 PRINT"{CLR}{RVS}*** LOAD ***
    { 2 GIU'}"                    :rem  212
795 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
    {OFF} PER USCIRE DAL LOAD{GIU'}"
                                  :rem   96
800 F$="":INPUT"{ 2 GIU'}NOME DEL FILE";F
    $:IFF$=""THENPRINT:GOTO310    :rem  115
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"    :rem  127
820 GETA$:IFA$<>"N"ANDA$<>"D"THEN820
                                  :rem   28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                  :rem  157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256              :rem    2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                  :rem  107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                            :rem   70
850 POKE780,0:SYS65493            :rem   11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                  :rem  111
865 PRINT"{GIU'}FATTO.":GOTO310 :rem  184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI!
    {GIU'}":IFDV=1THEN800         :rem   19
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800          :rem  102
1000 REM CICALINO                 :rem  231
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                          :rem  207
1002 POKE54276,33:POKE54273,6:POKE54272,5
                                  :rem   42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN     :rem  202
2000 REM CAMPANELLO               :rem  130
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                           :rem  152
2002 POKE54276,17:POKE54273,40:POKE54272,
     0                            :rem   86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                  :rem   57
3000 PRINTC$;"{RVS} NON IN PAGINA ZERO O
     SU{DES}ROM ":GOTO1000        :rem  240
```

FINALMENTE!

La Softrivista che ti gasa!



QUALCOSA DI SUPER, DI INEDITO, DI IRRESISTIBILE

IL VERO GIOCO COMINCIA ADESSO

IN EDICOLA JACKSON SOFT SERIE ORO

I giochi esclusivi per Commodore 64 e Spectrum 48 K importati dall'Inghilterra, mai presentati in Italia.
Una sfida Jackson al già visto, al già fatto, al... già registrato.





La prima puntata del fantastico, inedito PYJAMARAMA

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



GRUPPO EDITORIALE JACKSON

REM:HW

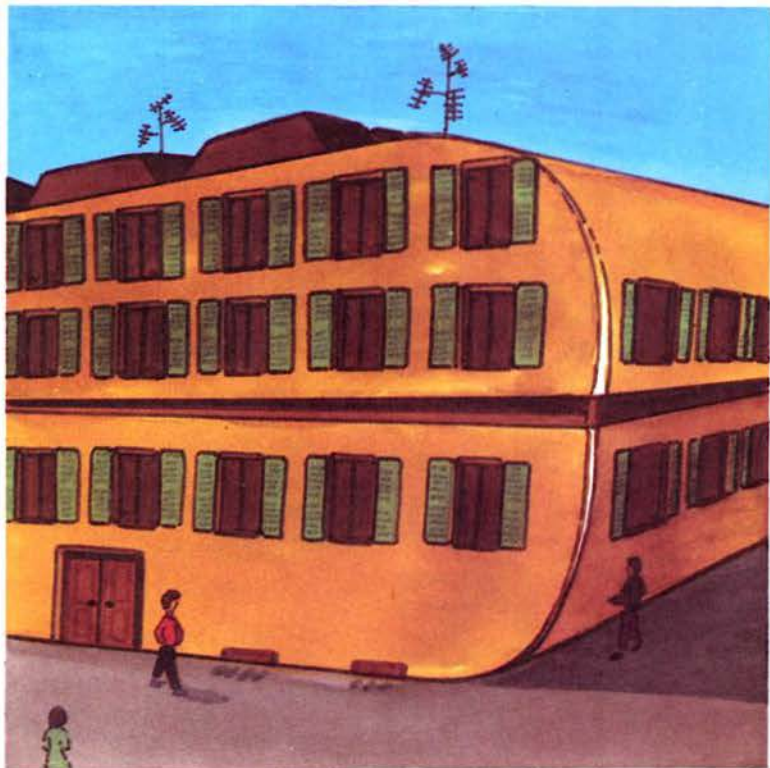
Hardware

di **M. Cristuib Grizzi**

Le possibilità di comunicazione con l'esterno dei computer Commodore

Il computer è una macchina per definizione interattiva: la sua potenza, oltre che nella velocità di calcolo e nell'ammontare di memoria disponibile, risiede soprattutto nelle sue capacità di comunicazione con il mondo esterno.

Quando digitiamo un programma, o diamo in modo diretto il comando LOAD per caricarlo in memoria da cassetta o disco, comunichiamo con il computer attraverso la pressione di alcuni tasti sulla tastiera: esso rileva la sequenza con la quale questi tasti vengono premuti, la confronta con il suo "vocabolario" interno e, se tutto è corretto, procede con l'esecuzione del comando che gli abbiamo così impartito. Se quanto abbiamo digitato non concorda con quanto il computer è in grado di capire, esso ci avvertirà con un "?SYNTAX ERROR", visualizzandolo sullo schermo.
La tastiera rappresenta il modo più immediato di "interfacciamento" uomo-computer, ed è il primo dispositivo di comunicazione del computer con l'esterno con il quale si prende confidenza. Il calcolatore, a seconda del lavoro che viene chiamato a svolgere, può tuttavia avere necessità di comunicare con altri apparati – ad esempio il registratore a cassette o la stampante – oppure ancora con l'uomo, attraverso altre "interfacce" che non siano la tastiera.



REM:HW

Per questi scopi il computer è dotato di un certo numero di prese – più propriamente "porte" – che permettono il collegamento con altri dispositivi.
Vediamo di analizzarle, per scoprire a che cosa servono.
La prima connessione del computer con l'esterno che l'utente impara a conoscere è senz'altro l'uscita video, la quale permette di collegare il computer al televisore, senza il quale esso rimarrebbe "muto", pur potendo ugualmente ed in ogni caso eseguire i suoi compiti.
Non si tratta propriamente di una porta, se si assume di definire con questo termine un canale di comunicazione con l'esterno in grado di permettere il trasferimento di dati. La connessione video è "a senso unico", in quanto permette soltanto l'uscita di un segnale dal computer e non invece il suo ingresso.
Questa uscita è divisa fisicamente in due diversi connettori, uno per il cavetto coassiale a 75 ohm da collegare direttamente alla presa di antenna di un comune televisore, e l'altro consistente in un connettore DIN per il collegamento ad un monitor a colori o monocromatico.
La differenza sostanziale tra un monitor ed un comune televisore a colori risiede nel trattamento al quale viene sottoposto il segnale proveniente dal computer: mentre nel televisore esso passa attraverso molti circuiti, che – tra le altre cose – lo sintonizzano sulla frequenza prescelta e ne separano i tre colori fondamentali (rosso, giallo e blu) prima di inviarli separatamente al tubo catodico, nel monitor il segnale entra già separato nelle sue tre componenti cromatiche fondamentali, non necessita di sintonia e viene inviato direttamente al tubo catodico. Per questo motivo i monitor dispongono di un segnale più "puro", permettendo un'immagine molto più stabile e definita rispetto a quella ottenibile anche dai migliori televisori a colori.
Il connettore DIN di collegamento al monitor è a otto poli nelle versioni di C64 più recenti (le prime versioni lo possedevano uguale a quello del VIC 20, a cinque poli), e possiede inoltre una linea di ingresso audio – oltre alla normale uscita – tale da permettere l'invio di segnali sonori dall'esterno direttamente all'integrato SID.
Chi volesse sperimentare con questo ingresso audio tenga presente che possono essere inviati al computer soltanto segnali di basso livello (non amplificati), onde evitare prevedibili guai all'integrato.
Dopo la connessione video, la porta – ed in questo caso il termine è appropriato – senz'altro più nota a tutti è quella di interfaccia con il registratore.
Questa porta, modificata nella forma, ma non nella sostanza, nei nuovi modelli C16 e Plus/4, permette di collegare il computer al Datassette 1530 (C2N) o, nel caso del C16-Plus/4, al Datassette 1531. I due modelli di registratore, a parte il colore esterno e qualche modifica non sostanziale nella circuiteria interna, sono del tutto identici e differiscono soltanto per la forma di questo connettore al computer.
Le connessioni utilizzate nella porta per il registratore comprendono una linea in grado di comunicare al computer se è stato premuto qualche tasto sul registratore stesso (qualche tasto, si noti bene, e non un tasto in particolare...), una linea per l'uscita dei dati dal computer ed una per il loro ingresso. Due linee sono utilizzate per l'alimentazione del registratore a due differenti tensioni: 9 V per il motore di trascinamento del nastro e 5 V per i circuiti elettronici interni, ed un'ultima linea viene utilizzata come massa.
La comunicazione attraverso questa porta viene completamente gestita dal computer attraverso il suo sistema operativo: il Kernal (questo è il nome assegnato dalla Commodore al sistema operativo delle sue macchine) possiede delle routine come la LOAD (C64: 65493), OPEN (C64: 65472), READST (C64: 65463), SAVE (C64: 65496), SETLFS (C64: 65466), SETNAM (C64: 65496), CLOSE (C64: 65475) ed altre, in grado di assicurare la completa gestione di questa periferica sia attraverso il BASIC che attraverso il linguaggio macchina. Una descrizione delle routine Kernal, per chi fosse interessato ad approfondire l'argomento, si può trovare sulla "Guida di riferimento del programmatore" edita dalla stessa Commodore per i vari computer (per C16 e Plus/4 non è attualmente ancora disponibile nemmeno in lingua inglese), oppure, nel caso del C64, in "Il S.O. del CBM64" della E.V.M. Computers.
Procedendo nella sequenza di analisi, dalle porte più conosciute a quelle meno conosciute, troviamo le (una sola nel VIC 20) Control port, meglio conosciute come porte joystick. Esse sono costituite da connettori a vaschetta DB-9 a nove poli, che permettono il collegamento della stragrande maggioranza di joystick presenti sul mercato. Anche qui la Commodore ha reso non standard queste porte nei nuovi C16 e Plus/4 – ufficialmente per diminuire l'interferenza RF di radiofrequenza (sic.) – costringendo l'acquirente delle nuove macchine all'acquisto di nuovi joystick... naturalmente Commodore.
Porte fondamentalmente di solo Input servono inoltre per il collegamento delle paddle (solo disponibili in coppia: consistono in due manopole rotative e due pulsanti di fuoco) e della penna ottica, che permette, dotata del software adeguato, di scrivere o disegnare direttamente sullo schermo. Nel C16 e Plus/4 non è invece presente la circuiteria necessaria alla lettura delle paddle, quindi queste porte vanno considerate come pure porte joystick.
La porta seriale DIN, sul retro del computer, viene usata comunemente per collegare il disk drive o la stampante.
Il termine "seriale" significa che i dati vengono trasmessi in serie, cioè un bit dopo l'altro. Una porta seriale può quindi funzionare solo in trasmissione o in ricezione, e non in entrambi i modi contemporaneamente, come una porta parallela.
Dotata di sei poli, deriva dalla porta seriale IEEE-488 presente nella prima serie PET, e comprende una linea cosiddetta di "reset" (la sua esistenza spiega il perché si accende la luce rossa del drive o si muove la testina della stampante ogni volta che il computer viene acceso). Come la porta del registratore, anche questa è controllata e gestita dal sistema operativo del calcolatore: le periferiche collegate sono distinte da un numero particolare (4 o 5 per la stampante, 6 per il plotter 1520, da 8 a 11 per il disk drive).
Questo numero distintivo della periferica può essere modificato via hardware e, in alcuni casi, anche via software (ad esempio, nel dischetto dimostrativo fornito nella confezione del drive è contenuto un programma che permette di modificare in 9 il numero di periferica distintivo del drive stesso). I comanti BASIC che accedono a questa porta sono i consueti OPEN, CLOSE, PRINT#, INPUT#, GET#, oltre al LOAD, SAVE e VERIFY sul disk drive, e da linguaggio macchina sono accessibili tutte le routine Kernal elencate per il registratore, insieme ad altre, come la LISTEN (C64: 65457), TALK (C64: 65460), CIOUT (C64: 65448), ACPTR (C64: 65445) e molte altre che non è possibile qui riportare per ragioni di spazio.
La User Port consiste invece nel connettore a 24 poli che si trova a destra, guardando il retro del computer. Come dice il suo stesso nome, è una porta disponibile per qualsiasi uso l'utente desidera farne: rende infatti possibile, attraverso i suoi connettori, l'accesso ad un gran numero di linee di controllo dell'integrato CIA (Complex Interface Adapter), o del VIA (Versatile Interface Adapter) nel caso del VIC 20; queste linee non sono tuttavia gestite dal sistema operativo, come nel caso della porta seriale o della porta per registratore, e richiedono quindi, per utilizzi personalizzati, un'attenta programmazione in linguaggio macchina.
A causa di queste limitazioni, la User

Port viene utilizzata , nella stragrande maggioranza dei casi, come porta di interfaccia RS-232. Dal momento tuttavia che il protocollo RS-232 usato dalla Commodore differisce da quello standard implementato da quasi tutte le periferiche RS-232, occorre connettere alla User Port un apposito adattatore – ad esempio l'adattatore VIC–1011A RS–232C, venduto dalla stessa Commodore – che permetta di convertire i livelli di tensione di 0.5 V della User Port nei livelli standard di −12 V e +12 V.

L'interfaccia RS-232 viene considerata dal computer come periferica numero 2, e parametri come velocità di trasmissione e parità vengono forniti, con una particolare codifica in CHR$, nel nome del file. L'interfaccia RS-232 è indispensabile, se si desiderano collegare al computer stampanti di qualità o modem non Commodore.

La User Port, esistente sul VIC 20, C64 e Plus/4, non è invece presente nel C16.

Infine, la porta che si trova a sinistra guardando il retro del computer viene utilizzata principalmente per espansioni di memoria (nel VIC 20) o per l'inserimento di cartucce di programmi. È collegata direttamente ad un buon numero di linee di controllo del micro processore, e questa particolarità rende possibile, ad esempio, il collegamento al C64 della cartuccia CP/M con il microprocessore Z80, che va a "sostituirsi" al 6510 presente nel computer stesso.

Ciò che è possibile ottenere con l'uso consapevole di tutte queste porte di comunicazione con il mondo esterno è limitato soltanto dalla disponibilità di accessori sul mercato, o dalla capacità dello sperimentatore hardware.

Negli USA già da tempo vengono commercializzati accessori molto particolari per il C64: da una serie di sensori, che, posti fuori dalla finestra di casa e collegati al computer, sono in grado di permettere al programma fornito insieme al tool i dati necessari per effettuare le previsioni del tempo, a connettori per inviare al computer le immagini riprese da una telecamera o contenute in una videocassetta, ai sintetizzatori di voce ed alle interfacce con strumenti musicali di tutti i tipi.





Astropanic per VIC 20

di **S.D.Wagle**
trad. ed adatt.
di **M. Cristuib Grizzi**

Un notevole programma completamente in linguaggio macchina per il VIC 20 inespanso, con caratteristiche di simulazione di SPRITE e movimento fine pixel per pixel, unitamente ad una grande velocità di esecuzione. Particolare curioso è l'età dell'autore del programma: 16 anni!



Come implicito nel nome, "Astropanic" è un gioco spaziale frenetico e dall'azione velocissima. Scopo del gioco, classicamente, è difendere – tramite l'uso del joystick – il proprio cannone nella parte bassa dello schermo dagli attacchi da parte delle innumerevoli astronavi aliene. Queste ultime sono raggruppate in stormi e si muovono in modo assolutamente imprevedibile, eseguendo delle veloci picchiate e risalite in quota.
Si dispone di tre cannoni, ed il gioco è strutturato su quindici diversi livelli, ognuno dei quali di difficoltà crescente. Se si dovesse superare il quindicesimo livello (ma non è molto facile ...) il programma continuerà ritornando al primo livello. L'azione può essere "congelata" in qualsiasi momento premendo il tasto SHIFT LOCK, che funziona come un interruttore, in quanto rilasciandolo si provocherà la ripresa dell'azione. Per iniziare, in qualsiasi momento, una nuova partita è sufficiente premere un qualunque tasto alfabetico.
Se si dispone della cassetta allegata alla rivista, è sufficiente posizionarsi con il nastro in corrispondenza dell'inizio del programma e digitare:

LOAD"",1,1

(attenzione: non inserire spazi tra gli apici!)
Il programma andrà quindi mandato in esecuzione digitando:

SYS 6560

Ricordiamo che per conoscere il punto esatto di inizio dei vari programmi su cassetta è sufficiente inserire quest'ultima completamente riavvolta nel registratore, azzerare il contanastro e digitare:

LOAD"*"

Dal momento che sicuramente non esiste sulla cassetta alcun programma di nome "*", il computer passerà in rassegna tutti i programmi incontrati, visualizzandone il nome con il messaggio:

FOUND XXXX

Sarà quindi utile annotarsi su un foglio i nomi dei programmi e la relativa posizione del contanastro (sottrarre tre o quattro unità dal valore corrispondente all'apparizzione del messaggio FOUND...).
Gli abbonati alla sola rivista dovranno invece usare il programma MLX (vedere nelle pagine verdi) per digitare il listato in linguaggio macchina. La procedura da seguire, in questo caso, è:
1. Spegnere il computer ed inserire un'espansione di memoria da almeno 8

Kbyte.
2. Accendere il computer e digitare:

POKE44,30:POKE7680,0:NEW
[RETURN]

Caricare in memoria MLX (che sarà stato precedentemente digitato e salvato), e rispondere alle sue richieste circa gli indirizzi iniziale e finale con:

indirizzo iniziale: 4608

indirizzo finale: 7679

Dopo avere digitato e salvato il programma tramite MLX si potranno seguire per il caricamento le istruzioni date per i possessori di cassetta.

Astropanic per VIC 20

```
4608 :162,000,189,074,003,201,117
4614 :094,144,003,076,106,019,192
4620 :234,234,234,234,189,070,183
4626 :003,016,021,010,133,174,119
4632 :056,189,068,003,229,174,231
4638 :157,068,003,176,020,222,164
4644 :066,003,076,055,018,234,232
4650 :010,024,125,068,003,157,173
4656 :068,003,144,003,254,066,074
4662 :003,134,174,234,189,066,086
4668 :003,201,001,144,007,201,105
4674 :166,176,059,076,192,018,241
4680 :032,148,224,166,174,165,213
4686 :141,157,070,003,169,127,233
4692 :157,071,003,165,143,048,159
4698 :024,234,234,234,165,142,099
4704 :041,128,029,071,003,157,013
4710 :071,003,094,070,003,169,000
4716 :001,157,066,003,076,192,091
4722 :018,094,070,003,094,071,208
4728 :003,076,091,018,234,234,008
4734 :234,234,032,148,224,166,140
4740 :174,165,141,157,070,003,074
4746 :169,127,157,071,003,165,062
4752 :143,048,029,169,128,094,243
4758 :070,003,029,070,003,157,226
4764 :070,003,165,142,041,128,193
4770 :029,071,003,157,071,003,240
4776 :169,166,157,066,003,076,037
4782 :192,018,094,070,003,094,133
4788 :071,003,076,147,018,234,217
4794 :234,234,234,234,234,234,054
4800 :189,071,003,016,021,010,246
4806 :133,174,056,189,069,003,054
4812 :229,174,157,069,003,176,244
4818 :020,222,067,003,076,231,061
4824 :018,234,010,024,125,069,184
4830 :003,157,069,003,144,003,089
4836 :254,067,003,134,174,234,070
4842 :189,067,003,201,003,144,073
4848 :007,201,150,176,059,076,141
4854 :106,019,032,148,224,166,173
4860 :174,165,141,157,071,003,195
4866 :169,127,157,070,003,165,181
4872 :143,048,024,234,234,234,157
4878 :165,142,041,128,029,070,077
4884 :003,157,070,003,094,071,162
4890 :003,169,003,157,067,003,172
4896 :076,106,019,094,071,003,145
4902 :094,070,003,076,014,019,058
4908 :234,234,234,234,032,148,136
4914 :224,166,174,165,141,157,053
4920 :071,003,169,127,157,070,141
4926 :003,165,143,048,029,169,107
4932 :128,094,071,003,029,071,208
4938 :003,157,071,003,165,142,103
4944 :041,128,029,070,003,157,252
4950 :070,003,169,150,157,067,190
4956 :003,076,106,019,094,071,205
4962 :003,094,070,003,076,067,155
4968 :019,234,024,138,105,012,124
4974 :041,252,170,224,083,176,032
4980 :003,076,002,018,076,234,013
4986 :019,234,234,234,162,000,237
4992 :032,238,025,189,067,003,170
4998 :074,074,074,168,185,000,197
5004 :027,157,072,003,185,032,104
5010 :027,157,073,003,189,066,149
5016 :003,074,074,074,024,125,014
5022 :072,003,157,072,003,144,097
5028 :003,254,073,003,024,138,147
5034 :105,012,041,252,170,224,206
5040 :083,176,003,076,128,019,149
5046 :234,076,000,020,234,234,212
5052 :234,234,234,234,189,064,097
5058 :003,133,176,189,065,003,251
5064 :133,177,169,032,160,000,103
5070 :145,176,200,145,176,160,184
5076 :022,145,176,200,145,176,052
5082 :076,016,020,189,074,003,084
5088 :201,094,144,220,134,178,171
5094 :076,016,020,234,238,192,238
5100 :003,173,192,003,205,193,237
5106 :003,176,003,076,000,018,006
5112 :169,000,141,192,003,076,061
5118 :126,019,032,208,025,162,058
5124 :000,142,063,003,234,234,168
5130 :234,234,234,076,221,019,004
5136 :189,074,003,201,000,208,179
5142 :100,173,063,003,010,010,125
5148 :010,010,010,133,179,024,138
5154 :189,067,003,041,007,101,186
5160 :179,133,179,105,008,133,009
5166 :177,134,178,189,066,003,025
5172 :041,006,010,010,010,170,043
5178 :164,179,189,032,029,153,036
5184 :000,028,189,040,029,153,247
5190 :016,028,232,200,196,177,151
5196 :208,238,238,063,003,024,082
5202 :165,178,105,012,041,252,067
5208 :170,224,083,144,176,162,023
5214 :000,189,074,003,201,000,049
5220 :240,009,201,088,176,005,051
5226 :169,144,141,169,003,138,102
```

```
5232 :024,105,012,170,224,083,218
5238 :144,231,076,128,020,076,025
5244 :000,023,234,234,162,000,009
5250 :142,063,003,134,178,189,071
5256 :072,003,133,176,189,073,014
5262 :003,133,177,173,063,003,182
5268 :010,010,133,187,160,000,136
5274 :177,176,133,188,032,000,092
5280 :021,160,022,177,176,133,081
5286 :188,230,187,032,000,021,056
5292 :160,001,177,176,133,188,239
5298 :230,187,032,000,021,160,040
5304 :023,177,176,133,188,230,087
5310 :187,032,000,021,166,178,006
5316 :138,024,105,012,170,238,115
5322 :063,003,224,083,144,181,132
5328 :173,141,002,201,000,240,197
5334 :008,169,001,141,194,003,218
5340 :076,208,020,169,000,141,066
5346 :194,003,234,169,014,141,213
5352 :015,144,173,169,003,141,109
5358 :013,144,169,007,141,248,192
5364 :151,076,160,022,234,234,097
5370 :234,234,234,234,234,234,118
5376 :132,189,165,187,010,010,181
5382 :010,170,105,008,133,190,110
5388 :165,188,201,031,176,019,024
5394 :010,010,010,168,189,000,149
5400 :028,025,000,028,157,000,006
5406 :028,232,200,228,190,208,092
5412 :241,164,189,166,178,189,139
5418 :074,003,201,094,144,001,047
5424 :096,165,187,145,176,096,145
5430 :234,234,234,234,234,234,178
5436 :234,234,234,234,173,160,049
5442 :003,201,001,240,074,169,242
5448 :001,141,160,003,173,060,098
5454 :003,141,162,003,160,000,035
5460 :162,060,041,007,201,003,046
5466 :144,007,160,001,162,061,113
5472 :234,234,234,140,166,003,083
5478 :142,161,003,173,060,003,132
5484 :074,074,074,024,105,184,131
5490 :141,164,003,169,031,141,251
5496 :165,003,169,160,141,163,153
5502 :003,234,173,060,003,041,128
5508 :007,024,109,162,003,141,066
5514 :162,003,169,240,141,012,097
5520 :144,076,045,024,234,234,133
5526 :234,234,234,234,234,234,018
5532 :234,234,234,234,173,160,145
5538 :003,201,001,208,073,162,042
5544 :000,189,066,003,056,233,203
5550 :004,205,162,003,176,044,000
5556 :105,012,205,162,003,144,043
5562 :037,173,163,003,024,105,179
5568 :008,221,067,003,144,026,149
5574 :173,163,003,056,233,015,073
5580 :221,067,003,176,015,189,107
5586 :074,003,201,000,208,008,192
5592 :076,160,023,234,234,234,153
5598 :234,234,138,024,105,012,201
5604 :170,224,083,144,192,234,251
5610 :234,234,234,234,234,234,102
5616 :096,234,234,234,234,234,226
5622 :234,234,234,234,234,234,114
5628 :234,234,234,234,234,234,120
5634 :234,234,234,234,234,234,126
5640 :234,234,234,234,234,234,132
5646 :234,234,173,160,003,201,251
5652 :001,208,105,172,166,003,163
5658 :173,164,003,133,251,173,155
5664 :165,003,133,252,169,032,018
5670 :145,251,056,173,164,003,062
5676 :233,022,176,003,206,165,081
5682 :003,141,164,003,056,173,078
5688 :163,003,233,008,141,163,255
5694 :003,201,160,144,011,169,238
5700 :000,141,160,003,141,013,014
5706 :144,076,191,234,173,164,032
5712 :003,133,251,173,165,003,040
5718 :133,252,172,166,003,173,217
5724 :161,003,133,254,032,160,067
5730 :021,165,254,145,251,173,083
5736 :160,003,201,001,208,008,173
5742 :173,012,144,233,007,141,052
5748 :012,144,234,234,234,234,184
5754 :234,234,234,234,234,234,246
5760 :173,005,144,201,253,208,088
5766 :005,169,255,141,005,144,085
5772 :169,014,173,160,003,201,092
5778 :001,240,005,169,096,141,030
5784 :012,144,076,191,234,234,019
5790 :234,234,162,004,189,068,025
5796 :027,024,125,074,027,201,130
5802 :010,144,008,254,073,027,174
5808 :233,010,076,169,022,157,075
5814 :074,027,169,000,157,068,165
5820 :027,189,074,027,024,105,122
5826 :176,157,234,031,202,224,194
5832 :255,208,215,162,003,160,179
5838 :128,136,192,255,208,251,096
5844 :202,224,255,208,244,173,238
5850 :195,003,024,105,176,141,094
5856 :249,031,173,193,003,201,050
5862 :010,160,176,144,004,200,156
5868 :056,233,010,140,245,031,183
5874 :024,105,176,141,246,031,197
5880 :076,128,026,234,234,234,156
5886 :234,234,254,074,003,254,027
5892 :074,003,234,234,234,189,204
5898 :074,003,041,240,201,096,153
5904 :144,005,076,058,023,234,044
5910 :234,134,178,032,044,023,155
5916 :010,010,010,010,010,133,211
5922 :179,189,067,003,041,007,008
5928 :076,069,023,234,024,105,059
```

LOAD

Astropanic per VIC 20

```
5934 :032,133,253,234,234,234,142
5940 :173,063,003,096,234,234,087
5946 :169,096,157,074,003,141,186
5952 :169,003,076,023,023,024,126
5958 :101,179,133,179,105,008,007
5964 :133,177,169,000,141,169,097
5970 :003,166,253,076,058,020,146
5976 :234,234,234,234,234,234,212
5982 :234,234,162,000,169,000,125
5988 :157,082,027,169,048,157,228
5994 :250,027,232,224,005,208,028
6000 :241,238,096,026,169,014,128
6006 :141,015,144,162,060,169,041
6012 :000,157,000,003,232,224,228
6018 :248,144,248,056,076,051,185
6024 :025,234,234,234,234,234,051
6030 :234,234,173,194,003,201,157
6036 :001,240,003,076,000,024,236
6042 :141,012,144,076,191,234,184
6048 :169,032,133,254,169,016,165
6054 :157,074,003,169,000,141,198
6060 :160,003,141,012,144,157,021
6066 :070,003,157,071,003,169,139
6072 :144,141,013,144,238,170,010
6078 :003,189,067,003,074,074,088
6084 :024,105,006,201,010,144,174
6090 :008,233,010,238,071,027,021
6096 :076,199,023,109,072,027,202
6102 :141,072,027,076,224,021,007
6108 :234,234,234,234,234,234,088
6114 :234,234,234,234,234,234,094
6120 :234,234,234,234,234,234,100
6126 :234,234,234,234,234,234,106
6132 :234,234,234,234,234,234,112
6138 :234,234,234,234,234,234,118
6144 :173,060,003,074,074,074,202
6150 :170,169,127,141,034,145,024
6156 :173,031,145,041,016,208,114
6162 :006,206,060,003,206,060,047
6168 :003,173,032,145,041,128,034
6174 :208,006,238,060,003,238,015
6180 :060,003,173,031,145,041,233
6186 :032,240,027,173,060,003,065
6192 :201,000,208,005,169,002,121
6198 :141,060,003,173,060,003,238
6204 :201,166,208,005,169,164,205
6210 :141,060,003,076,192,024,050
6216 :076,064,021,234,234,234,167
6222 :234,234,173,060,003,041,055
6228 :006,170,160,015,185,016,124
6234 :029,153,224,028,136,192,084
6240 :255,208,245,234,234,234,226
6246 :224,000,240,038,160,015,011
6252 :185,224,028,074,153,224,228
6258 :028,144,008,185,232,028,227
6264 :009,128,153,232,028,136,038
6270 :192,255,208,234,202,076,013
6276 :102,024,234,234,234,234,170
6282 :234,234,234,234,234,234,006
6288 :173,060,003,074,074,074,090
6294 :170,169,028,157,184,031,121
6300 :234,234,169,029,157,185,140
6306 :031,169,255,141,034,145,169
6312 :234,234,234,234,234,234,036
6318 :173,060,003,141,061,003,103
6324 :201,254,208,005,169,000,249
6330 :141,060,003,076,000,022,232
6336 :173,060,003,205,061,003,185
6342 :240,200,173,061,003,074,181
6348 :074,074,170,169,032,157,112
6354 :184,031,157,185,031,076,106
6360 :080,024,234,234,234,076,074
6366 :229,026,169,144,141,020,183
6372 :003,169,023,141,021,003,076
6378 :169,159,141,014,144,169,006
6384 :255,141,005,144,162,000,179
6390 :169,000,157,068,027,232,131
6396 :224,014,208,248,169,003,094
6402 :141,195,003,169,147,032,177
6408 :210,255,169,000,141,193,208
6414 :003,169,080,141,060,003,214
6420 :162,000,169,003,157,000,255
6426 :151,169,012,157,228,150,125
6432 :157,000,150,232,224,000,027
6438 :208,238,173,096,026,201,212
6444 :000,208,003,076,096,023,194
6450 :024,176,168,234,162,000,046
6456 :189,096,027,157,228,031,016
6462 :232,224,022,208,245,096,065
6468 :234,234,162,000,189,082,201
6474 :027,221,074,027,240,005,156
6480 :176,030,076,095,025,232,202
6486 :224,005,208,238,076,112,181
6492 :025,162,000,189,074,027,057
6498 :157,082,027,024,105,048,029
6504 :157,250,027,232,224,005,231
6510 :208,239,162,000,169,000,120
6516 :189,128,027,032,210,255,189
6522 :232,224,128,208,245,096,231
6528 :162,000,160,002,152,157,249
6534 :067,003,169,000,141,013,015
6540 :144,157,074,003,157,075,238
6546 :003,138,157,066,003,024,025
6552 :105,012,170,224,083,144,122
6558 :229,096,162,000,169,029,075
6564 :133,177,169,234,133,176,162
6570 :165,176,024,105,022,133,027
6576 :176,144,002,230,177,157,038
6582 :255,026,165,177,157,031,225
```



Astropanic per VIC 20

```
6588 :027,232,224,031,144,232,054
6594 :032,224,024,169,000,141,016
6600 :160,003,032,128,025,076,112
6606 :000,018,162,000,138,157,169
6612 :000,028,157,032,028,157,102
6618 :064,028,157,096,028,157,236
6624 :128,028,157,160,028,157,114
6630 :192,028,232,224,032,144,058
6636 :230,096,189,072,003,157,215
6642 :064,003,189,073,003,157,219
6648 :065,003,096,234,234,234,090
6654 :234,234,173,170,003,201,245
6660 :007,176,003,076,074,026,110
6666 :162,000,169,032,157,000,018
6672 :030,157,000,031,232,224,178
6678 :000,208,245,169,000,141,017
6684 :192,003,024,173,193,003,104
6690 :105,001,041,015,141,193,018
6696 :003,169,000,141,192,003,036
6702 :141,170,003,141,169,003,161
6708 :032,128,025,032,054,025,092
6714 :076,000,018,000,255,255,150
6720 :255,255,255,250,255,152,206
6726 :255,085,255,255,165,203,008
6732 :201,064,208,003,076,000,116
6738 :018,076,160,025,020,254,123
6744 :255,000,087,255,085,084,086
6750 :000,191,000,000,000,255,028
6756 :000,255,000,255,255,016,113
6762 :170,255,084,085,136,067,135
6768 :190,255,000,255,000,255,043
6774 :000,000,064,255,000,255,180
6780 :000,000,255,255,162,000,028
6786 :189,067,003,201,148,144,114
6792 :022,189,066,003,024,105,033
6798 :003,205,061,003,144,011,057
6804 :056,233,007,205,061,003,201
6810 :176,003,076,173,026,138,234
6816 :024,105,012,170,224,083,010
6822 :144,218,076,000,026,234,096
6828 :234,206,193,003,169,007,216
6834 :141,170,003,206,195,003,128
6840 :173,195,003,201,000,240,228
6846 :038,169,001,141,194,003,224
6852 :141,013,144,160,255,162,047
6858 :001,140,012,144,140,015,142
6864 :144,202,224,001,208,251,214
6870 :136,192,064,208,238,169,197
6876 :000,141,194,003,076,000,122
6882 :026,234,234,169,000,141,006
6888 :014,144,141,169,003,141,076
6894 :192,003,238,194,003,032,132
6900 :070,025,165,203,201,064,204
6906 :240,250,076,160,025,234,211
6912 :234,234,234,234,234,234,124
6918 :234,234,234,234,234,234,130
6924 :234,234,234,234,234,234,136
6930 :234,234,234,234,234,234,142
6936 :234,234,234,234,234,234,148
6942 :234,234,234,234,234,234,154
6948 :234,234,234,234,234,234,160
6954 :234,234,234,234,234,234,166
6960 :234,234,234,234,234,234,172
6966 :234,234,234,234,234,234,178
6972 :234,234,234,234,234,234,184
6978 :234,234,234,234,234,234,190
6984 :234,234,000,002,000,003,033
6990 :002,234,000,000,000,002,060
6996 :009,004,002,234,234,234,033
7002 :234,234,234,234,234,234,214
7008 :147,131,143,146,133,189,217
7014 :176,176,176,176,176,160,118
7020 :151,129,150,133,189,176,012
7026 :176,160,033,176,032,032,211
7032 :234,234,234,234,234,234,244
7038 :234,234,005,018,065,032,202
7044 :083,032,084,032,082,032,221
7050 :079,032,045,032,080,032,182
7056 :065,032,078,032,073,032,200
7062 :067,013,032,032,032,018,088
7068 :156,070,079,082,032,084,147
7074 :072,069,032,086,073,067,049
7080 :032,050,048,013,032,032,119
7086 :158,032,032,032,032,032,236
7092 :032,032,032,032,032,032,116
7098 :032,032,032,032,032,032,122
7104 :013,032,032,032,032,032,109
7110 :159,018,032,032,032,032,247
7116 :032,032,032,032,032,013,121
7122 :032,018,030,032,032,080,178
7128 :082,069,083,083,032,065,118
7134 :078,089,032,075,069,089,142
7140 :013,031,032,018,032,072,170
7146 :073,071,072,032,083,067,120
7152 :079,082,069,058,005,000,021
7158 :000,000,000,000,048,048,086
7164 :048,050,051,000,255,084,228
7170 :000,255,255,255,255,000,254
7176 :255,255,255,255,255,255,002
7182 :255,250,255,255,255,212,216
7188 :168,255,000,255,000,255,185
7194 :000,255,255,251,017,000,036
7200 :255,255,255,255,255,000,027
7206 :255,255,255,191,246,255,215
7212 :255,250,251,255,255,083,113
7218 :255,166,000,252,255,255,209
7224 :255,000,000,255,255,000,053
7230 :255,255,168,255,000,000,227
7236 :255,170,255,000,255,255,234
```



Astropanic per VIC 20

```
7242 :162,004,255,255,255,255,236
7248 :255,255,255,000,255,255,075
7254 :255,255,255,255,255,255,080
7260 :255,255,004,191,000,255,028
7266 :255,255,255,255,255,255,092
7272 :255,255,000,128,255,255,228
7278 :255,255,191,085,255,255,126
7284 :000,238,000,255,230,171,242
7290 :255,255,168,255,255,255,029
7296 :000,008,252,004,000,000,136
7302 :255,000,255,000,255,187,062
7308 :000,000,220,000,000,255,103
7314 :255,020,008,020,000,000,193
7320 :000,000,000,000,000,064,216
7326 :000,000,020,000,084,255,005
7332 :000,004,255,255,000,000,166
7338 :000,000,000,136,000,000,050
7344 :000,000,000,020,000,000,196
7350 :000,000,000,020,000,000,202
7356 :000,064,000,000,000,000,252
7362 :255,255,020,000,000,000,212
7368 :129,000,255,255,068,255,138
7374 :238,239,000,136,168,255,218
7380 :255,000,000,000,132,255,086
7386 :000,255,255,016,220,040,236
7392 :000,000,000,255,000,084,051
7398 :000,000,008,128,000,020,130
7404 :034,000,000,085,000,000,099
7410 :000,255,000,213,213,064,219
7416 :000,008,255,255,213,000,211
7422 :255,255,000,000,000,000,252
7428 :000,000,000,000,000,008,012
7434 :028,062,062,042,000,000,204
7440 :012,012,063,063,251,217,122
7446 :217,200,000,000,000,000,183
7452 :192,192,192,192,012,063,103
7458 :085,136,136,085,063,012,039
7464 :000,000,064,128,128,064,168
7470 :000,000,003,015,021,034,119
7476 :034,021,015,003,000,192,061
7482 :080,032,032,080,192,000,218
7488 :000,003,005,008,008,005,093
7494 :003,000,192,240,084,136,213
7500 :136,084,240,192,000,000,216
7506 :001,002,002,001,000,000,088
7512 :048,252,085,034,034,085,114
7518 :252,048,160,042,043,175,046
7524 :047,171,042,008,160,168,184
7530 :234,248,250,224,168,032,238
7536 :130,034,042,175,043,010,034
7542 :042,136,002,136,160,170,252
7548 :232,170,168,034,032,002,250
7554 :136,043,142,034,128,034,135
7560 :130,008,136,176,130,032,236
7566 :044,002,032,003,000,003,226
7572 :162,000,048,008,032,000,142
7578 :002,136,000,050,128,004,218
7584 :000,000,000,000,000,000,160
7590 :000,000,000,000,000,000,166
7596 :000,000,000,000,000,000,172
7602 :000,020,000,000,000,000,198
7608 :000,004,000,000,000,064,252
7614 :000,000,000,000,245,255,178
7620 :004,000,000,000,129,000,073
7626 :255,255,068,255,174,239,168
7632 :000,000,004,255,253,000,208
7638 :000,000,164,255,000,255,120
7644 :255,016,149,032,002,001,163
7650 :002,001,002,001,002,001,235
7656 :192,128,192,128,192,128,168
7662 :192,128,000,000,000,255,045
7668 :000,213,000,000,000,000,201
7674 :000,000,000,000,000,000,250
```





Un piccolo database

di **R. D. Gunn**
trad. ed adatt.
di **M. Cristuib Grizzi**

Questo breve programma costituisce un piccolo database per l'archiviazione di indirizzi, molto facile da usare ed estremamente efficiente. Funziona sia sul VIC 20 che sul C64, e può usare come memoria di massa sia il floppy disk che la cassetta a nastro magnetico.



Se collezionate foglietti di carta pieni di indirizzi e di numeri di telefono, o se avete una rubrica indirizzi che sta diventando sempre più illeggibile mano a mano che aggiornate gli indirizzi stessi o eliminate i più vecchi, è ora di cambiare.

Questo programma permette un accesso istantaneo ai nomi, indirizzi e numeri di telefono. Offre inoltre alcune caratteristiche speciali, come l'ordinamento alfabetico e la ricerca automatica di un nome nell'archivio.

Aggiungere dei record al file o "editare" il file stesso è molto semplice, dal momento che il programma accetta i comandi attraverso un menu e la risposta alle domande del computer è tutto ciò che viene richiesto all'utente.

Il programma è "unificato" e funziona sia sul C64 che sul VIC 20 con o senza espansione di memoria.

Scelte molto semplici

Dopo aver caricato il programma dategli il RUN. Apparirà il menu principale:

1. Carica file
2. Agg./Iniz.file
3. Edit file
4. Visualizza file
5. Save e fine

Per creare il file indirizzi selezionate l'opzione 2. Il numero del record (#1, se state iniziando un nuovo file) appare nella parte alta dello schermo. Vi viene quindi richiesto il cognome, il nome, la via (non digitate la virgola prima del numero civico, in quanto viene usata una normale istruzione di INPUT), la città, il CAP ed il numero di telefono.

Se decidete di non inserire dati in un campo (ad esempio, non sapete l'indirizzo di una persona), premete semplicemente RETURN. Alla fine degli inserimenti per ogni record, potrete scegliere tra "Successivo" (cioé continuare nell'inserimento dati), "Cancella" (il record appena digitato non viene inserito in memoria) o "Fine" (per ritornare al menu iniziale).

Quando avete finito di inserire i dati che vi servono selezionate l'opzione 5 per salvare il file. Il programma vi chiederà se volete usare il disco o la cassetta, ed il file verrà quindi salvato sulla memoria di massa. Chi usa il registratore dovrà prevedere un'unica cassetta contenente, come prima registrazione, il file dati registrato da questo programma.

Dopo che il file è stato creato e salvato disponete dei dati registrati in modo permanente su un supporto magnetico. Quando darete ancora il RUN al programma potrete quindi usare l'opzione 1 per ricaricare il file in memoria centrale. Per aggiungere nuovi record al file

usate ancora l'opzione 2: il numero di record visualizzato nella parte alta dello schermo sarà pari al numero di record già esistenti nel file, più uno.
Se volete operare un edit del file (ad esempio per modificare un numero di telefono), selezionate l'opzione 3: il programma vi chiederà di digitare il cognome della persona interessata. Verranno visualizzate tutte le informazioni concernenti quella persona, e quindi potrete scegliere cosa modificare dal seguente menu:

1. Cognome
2. Nome
3. Via
4. Città
5. CAP
6. Telefono
7. Cancella

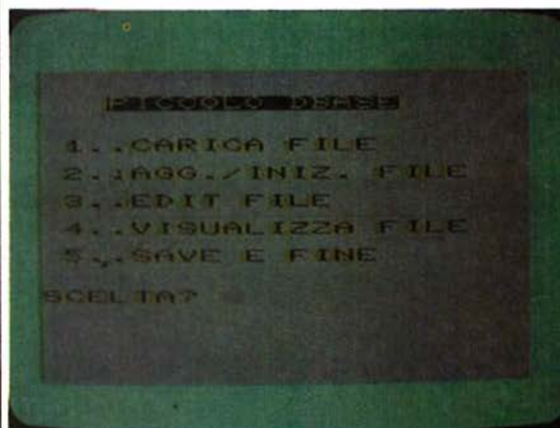
Selezionando una delle prime sei opzioni, verrà visualizzato il contenuto del campo relativo ed il programma chiederà di inserirne il contenuto corretto.
L'opzione 7 permette invece di cancellare il record dal file.
L'opzione 4 del menu principale richiama invece un altro submenu, in cui potrete scegliere tra il vedere un elenco alfabetico di tutti i nomi contenuti nel file, un nome individuale, oppure ritornare al menu principale.

Il numero dei record

Il programma è stato scritto in modo da permettere 12 record nel VIC inespanso (purtroppo la sua memoria è quella che è...) e 200 nel C64. Per aumentare il numero dei record incrementate il valore della variabile L in linea 10.
Naturalmente, se non inserite delle espansioni di memoria e modificate questo valore, otterrete prima o poi un inevitabile "?OUT OF MEMORY ERROR", con perdita di tutti i dati inseriti. Chi possiede il C64 deve notare che può essere incrementato solo il secondo valore di L (al termine della linea 10).

Un piccolo database
versione per VIC 20

```
10 L=12:SYS65517:A=PEEK(781):Q$=CHR$(13):
   IFA=40THENPOKE53281,15:L=200 :rem 241
20 DIMM$(L,6),N$(6):FORI=1TO6:READN$(I):N
   EXT                          :rem  93
30 DATACOGNOME,NOME,VIA,CITTA',CAP,TELEFO
   NO                           :rem  60
40 PRINT"{CLR}{ 2 GIU'}{ 3 DES}{RVS}{BLK}
   PICCOLO DBASE"Q$"{ 2 GIU'}{DES}1..CARI
   CA FILE"Q$"{GIU'}{DES}2..AGG./INIZ. FI
   LE"                          :rem 188
50 PRINT"{GIU'}{DES}3..EDIT FILE"Q$"
   {GIU'}{DES}4..VISUALIZZA FILE"Q$"
   {GIU'}{DES}5..SAVE E FINE"   :rem 240
60 INPUT"{ 2 GIU'}SCELTA";B:ONBGOSUB450,7
   0,130,240,450:GOTO40         :rem 120
70 N=N+1:PRINT"{CLR}RECORD #"N:FORJ=1TO6:
   PRINTQ$"{DES}{RVS}{BLK}";N$(J);"{OFF}
   {BLU}":INPUTM$(N,J)          :rem  35
80 NEXTJ:PRINT"{GIU'} {RVS}S{OFF}UCCESSIV
   O {RVS}C{OFF}ANCELLA {RVS}F{OFF}INE"
                                :rem  85
90 GETA$:IFA$=""OR(A$<>"S"ANDA$<>"C"ANDA$
   <>"F")THEN90                 :rem 210
100 IFA$="S"THEN70              :rem 239
110 IFA$="F"THEN40              :rem 224
120 N=N-1:GOTO70                :rem 164
130 GOSUB660:FORI=1TON:IFB$=M$(I,1)THEN15
    0                           :rem  18
140 NEXTI:PRINT"{CLR}{GIU'}"B$"{GIU'}NON
    E'NEL FILE.":GOSUB620:GOTO40:rem  52
150 GOSUB650:PRINT"{GIU'}COSA MODIFICHI:"
    :FORJ=1TO6:PRINT"{DES}"J"..",N$(J):NEX
    T                           :rem   3
160 PRINT"{ 2 DES}7.. CANCELLA{GIU'}":INP
    UT"COSA SCEGLI";J:IFJ<1ORJ>7THEN160
                                :rem 143
170 IFJ=7THEN210                :rem 167
180 PRINT"{CLR}{RVS}"M$(I,1)", "M$(I,2)
                                :rem  80
190 PRINT"{GIU'}{RVS}{BLK}VECCHIO "N$(J)"
     E'{BLU}":PRINTM$(I,J)Q$Q$"{RVS}{BLK}
    CORRETTO "N$(J)" E'{BLU}"   :rem  97
200 INPUTM$(I,J):GOSUB620:GOTO220
                                :rem 252
210 FORA=ITON-1:FORJ=1TO6:M$(A,J)=M$(A+1,
    J):NEXTJ,A:N=N-1            :rem 168
220 PRINT"{CLR}{GIU'}{DES}VUOI:"Q$"{GIU'}
    {DES}1..CONTINUARE EDIT"Q$"{GIU'}
    {DES}2..RITORNARE AL MENU{ 2 GIU'}"
                                :rem  68
230 INPUT"{DES}COSA SCEGLI";B:ONBGOTO130,
    40:GOTO230                  :rem  94
240 PRINT"{CLR}{BLK}{GIU'}QUALE VUOI?"Q$"
    {GIU'}{DES}1..ALFABETICA"Q$"{GIU'}
    {DES}2..INDIVIDUALE"        :rem 104
250 PRINT"{GIU'}{DES}3..MENU{ 2 GIU'}":IN
    PUT"COSA SCEGLI";B:ONBGOTO260,410,40:
    GOTO250                     :rem  41
260 G=N:IFN=0THENPRINT"{GIU'}IL FILE E' V
    UOTO!":FORI=1TO2000:NEXT:GOTO40
                                :rem 107
270 G=INT(G/2)                  :rem   1
280 IFG=0THEN380                :rem 167
290 FORA=1TON-G                 :rem 150
300 IFM$(A,1)<=M$(A+G,1)THEN350 :rem 148
310 R$=M$(A,1):S$=M$(A,2):T$=M$(A,3):U$=M
    $(A,4):V$=M$(A,5):W$=M$(A,6):rem  70
320 FORJ=1TO6:M$(A,J)=M$(A+G,J):NEXT
                                :rem  97
330 M$(A+G,1)=R$:M$(A+G,2)=S$:M$(A+G,3)=T
    $:M$(A+G,4)=U$:M$(A+G,5)=V$ :rem  43
340 M$(A+G,6)=W$:E=A            :rem  35
350 NEXTA                       :rem  24
360 IFE=0THEN270                :rem 162
370 E=0:GOTO280                 :rem  89
380 PRINT"{CLR}":L=0:FORI=1TON:L=L+1:PRIN
    TM$(I,1)", "M$(I,2):IFL<>10THEN400
                                :rem 225
390 L=0:GOSUB620:PRINT"{CLR}"   :rem  69
400 NEXTI:GOSUB620:GOTO240      :rem 119
410 GOSUB660:FORI=1TON:IFB$=M$(I,1)THEN44
    0                           :rem  21
```





LOAD

Un piccolo database

```
420 NEXTI                              :rem  30
430 PRINT"{CLR}{ 2 GIU'}";B$:PRINT"{GIU'}
    NON E'NEL FILE":GOSUB620:GOTO240
                                       :rem 139
440 GOSUB650:GOSUB620:GOTO240          :rem  14
450 PRINT"{CLR}DISCO O CASSETTA (D/C)"
                                       :rem  29
460 GETE$:IF(E$<>"D"ANDE$<>"C")ORE$=""THE
    N460                               :rem 249
470 IFE$="D"THENF$="@0:INDIRIZZI":F=1:GOT
    O490                               :rem 214
480 F$="INDIRIZZI":F=0:G$=""           :rem  86
490 IFB=5THEN540                       :rem 168
500 IFF=1THENG$=",S,R"                 :rem 240
510 OPEN1,1+7*F,8*F,F$+G$:GOSUB580
                                       :rem 127
520 INPUT#1,N:FORI=1TON:FORJ=1TO6
                                       :rem 112
530 INPUT#1,M$(I,J):NEXT:NEXT:GOSUB580:CL
    OSE1:GOSUB580:GOTO40               :rem 129
540 IFF=1THENG$=",S,W"                 :rem 249
550 OPEN1,1+7*F,1+7*F,F$+G$:GOSUB580
                                       :rem 222
560 PRINT#1,N:FORI=1TON:FORJ=1TO6
                                       :rem 113
570 PRINT#1,M$(I,J):NEXT:NEXT:GOSUB580:CL
    OSE1:GOSUB580:END                  :rem 188
580 IFF=0THENRETURN                    :rem 238
590 IFO=0THENOPEN15,8,15:O=1           :rem 157
600 INPUT#15,A,B$,C,D:IFATHENPRINTA,B$,C,
    D:STOP                             :rem 133
610 RETURN                             :rem 119
620 PRINT"{ 2 GIU'}{RVS}{BLK}UN TASTO PER
     CONTINUARE"                       :rem  58
630 GETA$:IFA$=""THEN630               :rem  85
640 RETURN                             :rem 122
650 PRINT"{CLR}{GIU'}"M$(I,1)", "M$(I,2)Q
    $M$(I,3)Q$M$(I,4)" "M$(I,5)Q$M$(I,6):
    RETURN                             :rem 188
660 PRINT"{CLR}{GIU'}{ 3 DES}QUALE RECORD
    "Q$"{ 3 DES}VUOI (COGNOME)?":INPUT"
    { 2 GIU'}";B$:RETURN               :rem 117
```

Un piccolo database
versione per C-16

```
    LOR1,2
110 DIMM$(L,6),N$(6):FORI=1TO6:READN$(I):
    NEXT
120 DATACOGNOME,NOME,VIA,CITTA',CAP,TELEF
    ONO
130 PRINT"{CLR}{ 2 GIU'}{ 3 DES}{RVS}AGEN
    DA TELEFONICA"Q$"{ 2 GIU'}{DES}1..CAR
    ICA FILE"Q$"{GIU'}{DES}2..AGG./INIZ.
    FILE"
140 PRINT"{GIU'}{DES}3..EDIT FILE"Q$"
    {GIU'}{DES}4..VISUALIZZA FILE"Q$"
    {GIU'}{DES}5..ELENCO COMPLETO"
150 PRINT"{GIU'}{DES}6..SAVE E FINE"
160 INPUT"{ 2 GIU'}SCELTA";B:ONBGOSUB550,
    170,230,340,770,550:GOTO130
170 N=N+1:PRINT"{CLR}RECORD #"N:FORJ=1TO6
    :PRINTQ$"{DES}{RVS}";N$(J);"{OFF}":IN
    PUTM$(N,J)
180 NEXTJ:PRINT"{GIU'} {RVS}S{OFF}UCCESSI
    VO {RVS}C{OFF}ANCELLA {RVS}F{OFF}INE"
190 GETA$:IFA$=""OR(A$<>"S"ANDA$<>"C"ANDA
    $<>"F")THEN190
200 IFA$="S"THEN170
210 IFA$="F"THEN130
220 N=N-1:GOTO170
230 GOSUB760:FORI=1TON:IFB$=M$(I,1)THEN25
    0
240 NEXTI:PRINT"{CLR}{GIU'}"B$"{GIU'}NON
    E'NEL FILE.":GOSUB720:GOTO130
250 GOSUB750:PRINT"{GIU'}COSA MODIFICHI:"
    :FORJ=1TO6:PRINT"{DES}"J"..”N$(J):NEX
    T
260 PRINT"{ 2 DES}7.. CANCELLA{GIU'}":INP
    UT"COSA SCEGLI";J:IFJ<1ORJ>7THEN260
270 IFJ=7THEN310
280 PRINT"{CLR}{RVS}"M$(I,1)", "M$(I,2)
290 PRINT"{GIU'}{RVS}VECCHIO "N$(J)" E'":
    PRINTM$(I,J)Q$Q$"{RVS}CORRETTO "N$(J)
    " E'"
300 INPUTM$(I,J):GOSUB720:GOTO320
310 FORA=ITON-1:FORJ=1TO6:M$(A,J)=M$(A+1,
    J):NEXTJ,A:N=N-1
320 PRINT"{CLR}{GIU'}{DES}VUOI:"Q$"{GIU'}
    {DES}1..CONTINUARE EDIT"Q$"{GIU'}
    {DES}2..RITORNARE AL MENU{ 2 GIU'}"
330 INPUT"{DES}COSA SCEGLI";B:ONBGOTO230,
    130:GOTO330
340 PRINT"{CLR}{GIU'}QUALE VUOI?"Q$"
    {GIU'}{DES}1..ALFABETICA"Q$"{GIU'}
    {DES}2..INDIVIDUALE"
350 PRINT"{GIU'}{DES}3..MENU{ 2 GIU'}":IN
    PUT"COSA SCEGLI";B:ONBGOTO360,510,130
    :GOTO350
360 G=N:IFN=0THENPRINT"{GIU'}IL FILE E' V
    UOTO!":FORI=1TO2000:NEXT:GOTO130
370 G=INT(G/2)
380 IFG=0THEN480
390 FORA=1TON-G
400 IFM$(A,1)<=M$(A+G,1)THEN450
410 R$=M$(A,1):S$=M$(A,2):T$=M$(A,3):U$=M
    $(A,4):V$=M$(A,5):W$=M$(A,6)
420 FORJ=1TO6:M$(A,J)=M$(A+G,J):NEXT
430 M$(A+G,1)=R$:M$(A+G,2)=S$:M$(A+G,3)=T
    $:M$(A+G,4)=U$:M$(A+G,5)=V$
440 M$(A+G,6)=W$:E=A
450 NEXTA
460 IFE=0THEN370
470 E=0:GOTO380
480 PRINT"{CLR}":L=0:FORI=1TON:L=L+1:PRIN
    TM$(I,1)", "M$(I,2):IFL<>10THEN500
490 L=0:GOSUB720:PRINT"{CLR}"
500 NEXTI:GOSUB720:GOTO340
510 GOSUB760:FORI=1TON:IFB$=M$(I,1)THEN54
    0
520 NEXTI
530 PRINT"{CLR}{ 2 GIU'}";B$:PRINT"{GIU'}
    NON E'NEL FILE":GOSUB720:GOTO340
540 GOSUB750:GOSUB720:GOTO340
550 PRINT"{CLR}DISCO O CASSETTA (D/C)"
560 GETE$:IF(E$<>"D"ANDE$<>"C")ORE$=""THE
    N560
570 IFE$="D"THENF$="@0:INDIRIZZI":F=1:GOT
    O590
580 F$="INDIRIZZI":F=0:G$=""
590 IFB=5THEN640
```

Un piccolo database

```
600 IFF=1THENG$=",S,R"
610 OPEN1,1+7*F,8*F,F$+G$:GOSUB680
620 INPUT#1,N:FORI=1TON:FORJ=1TO6
630 INPUT#1,M$(I,J):NEXT:NEXT:GOSUB680:CL
    OSE1:GOSUB680:GOTO130
640 IFF=1THENG$=",S,W"
650 OPEN1,1+7*F,1+7*F,F$+G$:GOSUB680
660 PRINT#1,N:FORI=1TON:FORJ=1TO6
670 PRINT#1,M$(I,J):NEXT:NEXT:GOSUB680:CL
    OSE1:GOSUB680:END
680 IFF=0THENRETURN
690 IFO=0THENOPEN15,8,15:O=1
700 INPUT#15,A,B$,C,D:IFATHENPRINTA,B$,C,
    D:STOP
710 RETURN
720 PRINT"{ 2 GIU'}{RVS}UN TASTO PER CONT
    INUARE"
730 GETA$:IFA$=""THEN730
740 RETURN
750 PRINT"{CLR}{GIU'}"M$(I,1)", "M$(I,2)Q
    $M$(I,3)Q$M$(I,4)" "M$(I,5)Q$M$(I,6):
    RETURN
760 PRINT"{CLR}{GIU'}{ 3 DES}QUALE RECORD
    "Q$"{ 3 DES}VUOI  (COGNOME)?":INPUT"
    { 2 GIU'}";B$:RETURN
770 IFN=0THENPRINT"{CLR}{GIU'}IL FILE E'
    VUOTO!!":FORI=0TO3000:NEXT:RETURN
780 FORI=1TON
790 PRINT"{CLR}RECORD #"I
800 FORJ=1TO6
810 PRINTQ$"{RVS}";N$(J);": ";M$(I,J)
820 NEXTJ
830 PRINT"{ 2 GIU'}UN TASTO PER PROSEGUIR
    E":PRINT"Q PER USCIRE DALL'OPZIONE"
840 GETKEY A$:IF A$="Q"THENRETURN
850 NEXT I
860 RETURN
```

1..CARICA FILE
2..ADD./INIZ. FILE
3..EDIT FILE
4..VISUALIZZA FILE
5..SAVE E FINE
SCELTA?

LIBRI FIRMATI JACKSON

NOVITÀ

CON CASSETTA

Gaetano Marano
77 PROGRAMMI PER SPECTRUM

Oltre alla qualità dei programmi contenuti in questo libro, è da notare la varietà delle applicazioni e degli argomenti, per la cui realizzazione, si è cercato di utilizzare al massimo le nuove interessanti prestazioni dello Spectrum quali: le nuove istruzioni BASIC, la velocità, il suono, e soprattutto il colore e la grafica.
Cod. 566A Pag. 150 ____ Lire. 24.000 con cassetta

Roberto Rigo
SPECTRUM TOOL.
Programmi di qualità: grafica e gioco

Questo libro propone una serie di interessanti programmi BASIC che si servono di routine scritte in linguaggio macchina. con cassetta
Cod. 5670 Pag. 186 ____ Lire. 24.000

Mark Ramshaw
GIOCHI GIOCHI GIOCHI PER IL VOSTRO VIC 20

I giochi: quanti hanno cominciato proprio da qui a interessarsi di Personal Computer?
Questo libro ne presenta diversi di diversa difficoltà: due parole su come si gioca e poi il listato del programma.
Cod. 568D Pag. 114 ____ Lire. 24.000 con cassetta

Tim Hartnell
SINFONIA PER UN COMPUTER: VIC 20

Questo libro è un invito a imparare divertendosi; prima giocate, ma poi date un'occhiata ai listati dei programmi. Imparerete alcune tecniche semplici ma raffinate, per raggiungere gli effetti più spettacolari, e queste vi serviranno senz'altro quando dai giochi vorrete passare ad applicazioni ... "serie".
Cod. 5650 Pag. 130 ____ Lire. 24.000 con cassetta

Owen Bishop
IL LIBRO DEI GIOCHI DEL COMMODORE 64

Ogni programma è corredato da istruzioni che ti insegnano l'uso dei giochi. Il modo in cui essi funzionano è esposto in dettaglio, con alcuni accorgimenti sulle tattiche vincenti. Vengono dati pure alcuni consigli su come modificare i giochi in forma creativa per adattarli alle tue idee.
Cod. 3500 Pag. 158 ____ Lire. 24.000 con cassetta

Nigel Freestone
DATABASE: L'UTILE E IL DILETTEVOLE

Il libro, studiato per chi vuol "fare da sé" con il calcolatore, parla di quella che è la materia prima dell'elaborazione, i dati, e affronta il problema della loro memorizzazione, della loro organizzazione all'interno del computer e delle operazioni che si possono fare su di essi.
Cod. 500H Pag. 160 ____ Lire. 15.000

GRUPPO EDITORIALE JACKSON

Francesco Lentini
CONOSCI TE STESSO TRAMITE IL TUO PERSONAL COMPUTER

Il libro si compone di cinque capitoli, ognuno dei quali attinge ad una delle discipline che affrontano in qualche modo lo studio della personalità umana: Numerologia, Astrologia, Grafologia, Scienza del Comportamento, Test di Intelligenza.
I programmi sono stati scritti per Apple II e sono disponibili su cassetta e dischetto.
Cod. 412D Pag. 128 Lire 24.000

Michel Benelfoul
GIOCHI, TRUCCHI E CONTI

Questo libro è destinato soprattutto ai programmatori principianti, i suoi 29 programmi, presentati con una ricca documentazione, sono pronti per essere eseguiti. Gli argomenti sono tanti e rispondono alle esigenze più disparate. Il BASIC è quello dei calcolatori PET/CBM di cui vengono sfruttate a pieno le molte funzioni.
Cod. 553D Pag. 164 ____ Lire. 24.000 con cassetta

W. Carris
BASIC ATARI

Un ottimo libro per chi non ha alcuna esperienza di programmazione e di computer, per chi sà tutto sul BASIC ma non sul Basic Atari e infine per i programmatori Atari che potranno usare il testo come manuale di riferimento.
Cod. 409D Pag. 192 Lire 18.000

LA BIBLIOTECA CHE FA TESTO

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON – Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con **esenzione del contributo spese di spedizione:**

☐ Allego assegno della Banca ____________

N° ____________

☐ **Allego fotocopia del versamento su c/c n. 11666203 a voi intestato**

☐ **Allego fotocopia di versamento su vaglia postale a voi intestato**

Nome ____________
Cognome ____________
Via ____________
Cap ______ Città ______ Prov. ______
Data ______ Firma ______

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I V A |_|_|_|_|_|_|_|_|_|_|_|

ORDINE MINIMO L. 50.000



Forbidden Crypt

di **T. Reynolds**
trad. e adatt.
di **S. Albarelli, M. Anticoli**
e **M. Cristuib Grizzi**

In questo gioco d'azione ci si potrà mettere nei panni di un archeologo in cerca di tesori nascosti. Questi, però, vengono sorvegliati da accanite creature e non è facile impadronirsene. Forbidden Crypt è un emozionante gioco per VIC 20 + 8 Kbyte, C64, C16 e Plus/4 provvisti di joystick!

Una leggenda del luogo in cui vi trovate parla di un grande e potente re che viveva, un tempo, nei paraggi.
Dopo aver investigato su questa persona si seppe che egli aveva nascosto nei sotterranei della sua antica dimora, il castello maledetto, un tesoro inestimabile, protetto da creature feroci.
Questa sensazionale scoperta destò l'attenzione di tutti gli archeologi del mondo, che però non ebbero mai il coraggio di affrontare la difficile esplorazione.
Perché non cercare di accaparrarsi i tesori custoditi nel castello maledetto e diventare l'archeologo più famoso del mondo?
Soppravvivere nei sotterranei del castello maledetto non è però cosa del tutto semplice. Ogni piano del sotterraneo è composto da quattro stanze, in ognuna delle quali è custodito un tesoro sorvegliato da creature feroci e accanite, come serpenti, scorpioni, scheletri e fantasmi.
L'unico modo per procedere nella discesa al piano sottostante, nel quale sono custoditi altri quattro inestimabili tesori, è quello di prendere tutti i tesori presenti al piano in cui ci si trova.
All'inizio del gioco si possono vedere dall'alto le quattro stanze del piano in cui ci si trova e, nella parte bassa dello schermo, il punteggio aggiornato, il numero di vite e il livello.
Il giocatore è rappresentato, in questa schermata, dalla pallina, che è possibile muovere mediante il joystick (porta 2 per il C64 e porta 1 per il C16 e il Plus/4).



Per entrare nel vivo dell'esplorazione è necessario far entrare la pallina in una porta di una delle quattro stanze.
Come si entra nella stanza prescelta si è immediatamente inseguiti dalle creature che difendono ferocemente il tesoro (i nemici per ogni stanza sono quattro).
Da esse ci si può difendere sparando mediante il tasto "fire" nella direzione in cui ci si muove.
Nella versione per C64 i proiettili disponibili per ogni stanza sono solo tre; invece in quelle per VIC 20, C16 e Plus/4 si possiede una scorta infinita di proiettili.

Se le creature riescono a toccare il giocatore, esso perde una delle sue vite.
Uscendo da una stanza si ritorna a vedere le quattro stanze dall'alto; in quelle in cui è già stato preso il tesoro ci sarà un quadratino.
Ricordiamo che per passare ai piani sottostanti del sotterraneo è necessario prendere tutti e quattro i tesori.
Nella versione per C16, C64 e Plus/4 all'inizio della partita si hanno a disposizione cinque vite e se ne acquista una per ogni quattro piani del sotterraneo superati.
Invece nella versione per VIC 20 all'inizio della partita si possiedono dieci vite,

ma non è possibile acquistarne altre.

Note alla versione per VIC 20

Nella versione per VIC 20 è necessaria come minimo l'espansione da 8 Kbyte (si possono utilizzare anche le espansioni da 16, 24 e 32 Kbyte).
Questa versione è divisa in tre programmi.
Il primo programma modifica il contenuto di alcuni puntatori della memoria, protegge una zona di essa per usi specifici e carica il secondo programma.
Questo ridefinisce alcuni caratteri e carica il terzo programma, che è il gioco vero e proprio.

Note alla versione per C16 e Plus/4

Il C16 e il Plus/4 sono compatibili e utilizzano lo stesso programma.
Una caratteristica molto utile che entrambi possiedono è la possibilità di ridefinire i tasti funzione.
Chi non possedesse la versione su cassetta fornita allegata alla rivista, e dovesse battere il programma, può risparmiarsi parecchia fatica programmando i tasti di funzione nel seguente modo:

F1.....POKE
F2.....PRINT
F3.....GOSUB
F4.....DATA
F5.....THEN
F6.....GOTO

È consigliabile non programmare il tasto F7, perché contiene la funzione HELP, che è molto utile per correggere eventuali errori di battitura.

Forbidden Crypt

versione per C64

```
100 PRINT"{CLR}{ 12 GIU'}{ 12 DES}ATTENDE
    RE..."                          :rem  67
101 SN=54272:FORI=SNTOSN+24:POKEI,0:NEXT:
    AD=54277:SR=AD+1:LF=SN:LH=SN+1
                                    :rem 204
102 CR=SN+4:VL=54296                :rem 136
103 POKEAD,16:POKESR,240            :rem 171
104 POKEVL,15                       :rem 248
110 GOSUB820:GOSUB1810              :rem  48
115 T$="{RVS}{ 40 SPAZI}{OFF}":P=5:SC=0:L
    V=1                             :rem 183
116 FORI=1TO4:RM(I)=0:NEXT          :rem 102
120 S$="{RVS}{ 2 SPAZI}{OFF}{ 36 SPAZI}
    {RVS}{ 2 SPAZI}{OFF}"           :rem 211
125 POKE49194,7                     :rem  53
130 DEFFNRX(X)=INT(RND(1)*87)+X:DEFFNRY(Y
    )=INT(RND(1)*57)+Y:POKE53280,0
                                    :rem 247
140 POKE53281,0:PL=1524:PX=20:PY=12:OPL=1
    524:POKEPL,81:POKEPL+54272,4:EN=53269
                                    :rem 117
150 POKEEN,255:POKE49192,0:LV=1:SC=0:CS=5
    3278:CD=53279:GOSUB330:GOTO115
                                    :rem 237
160 PRINT"{CLR}";:PRINT:PRINTT$;S$;S$;:GO
    SUB180:PRINTS$;S$;:GOSUB180:PRINTS$;S
    $;T$                            :rem  79
170 PRINT"{HOME}":RETURN            :rem 150
180 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    [<A>]{ 13 *}[<S>]{ 2 SPAZI}[<A>]
    { 13 *}[<S>]{ 2 SPAZI}{RVS}
    { 2 SPAZI}{OFF}";               :rem  41
190 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
                                    :rem  98
200 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
                                    :rem  90
210 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    I{ 13 SPAZI}I{ 2 SPAZI}I{ 13 SPAZI}I
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
                                    :rem  11
220 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
                                    :rem  92
230 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
                                    :rem  93
240 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    [<Z>]{ 13 *}[<X>]{ 2 SPAZI}[<Z>]
    { 13 *}[<X>]{ 2 SPAZI}{RVS}
    { 2 SPAZI}{OFF}";               :rem  62
250 RETURN                          :rem 119
260 PRINT"{CLR}";:PRINT:PRINTT$;:rem 120
270 FORI=1TO20:PRINTS$;:NEXT:PRINTT$:PRIN
    T"{HOME}":PRINT"{ 9 GIU'}"; :rem 106
280 D$="[<L>][<J>]{ 39 SPAZI}[<L>][<J>]":
    PRINTD$;D$;D$;D$:RETURN         :rem 139
290 PRINT"{HOME}{ 23 GIU'}{YEL}
    { 2 SPAZI}PUNTI"SC"{ 3 SPAZI}LIV. "LV
    "{ 3 SPAZI}VITE "P"[<7>]":      :rem 173
291 PRINT"{HOME}":RETURN            :rem 154
330 IFTC<>4THEN340                  :rem  48
335 TC=0:FORI=1TO4:RM(I)=0:NEXT:LV=LV+1:G
    OSUB880                         :rem  24
336 IFLV/4=INT(LV/4)THENP=P+1:V=7-INT(LV/
    4):IFV>=0THENPOKE49194,V        :rem  61
340 POKE49192,0:POKE53269,0:GOSUB160:GOSU
    B290:IFP=0THEN610               :rem 235
341 IFRM(1)=1THENPOKE1315,160       :rem 161
342 IFRM(2)=1THENPOKE1332,160       :rem 162
343 IFRM(3)=1THENPOKE1675,160       :rem 174
344 IFRM(4)=1THENPOKE1692,160       :rem 175
350 POKEPL,81:POKEPL+54272,4        :rem 140
360 M=PEEK(56320):M=(MAND15)        :rem  52
370 IFM=15THEN360                   :rem 225
380 DX=((M>8)AND(M<12))-(M<8)       :rem  40
390 DY=((M=6)OR(M=10)OR(M=14))-((M=5)OR(M
    =9)OR(M=13))                    :rem 185
400 PX=PX+DX:PY=PY+DY               :rem 121
410 PL=(PY*40)+1024+PX:IFPEEK(PL)<>32THEN
    430                             :rem  59
420 POKEOPL,32:POKEOPL+54272,0:POKEPL,81:
    POKEPL+54272,4:OPL=PL:GOTO360
                                    :rem  88
430 IFPEEK(PL)<>9THENGOTO590        :rem 241
440 IFPY=7THENRM=1                  :rem 128
450 IFPY=16THENRM=3                 :rem 179
460 IFPX>19THENRM=RM+1              :rem 127
470 IFRM=1THENCL=6                  :rem 104
480 IFRM=2THENCL=7                  :rem 107
490 IFRM=3THENCL=5                  :rem 107
500 IFRM=4THENCL=2                  :rem  97
510 POKE53292,CL                    :rem 133
520 IFRM(RM)=1THENPX=PX-DX:PY=PY-DY:PL=(P
    Y*40)+1024+PX:OPL=PL:GOTO420:rem 162
530 FORI=1TO4:POKE2040+I,230+RM:POKE53287
    +I,CL+1:NEXT:POKE2045,235+RM:GOSUB650
                                    :rem 202
540 X=PEEK(53248):Y=PEEK(53249) :rem 247
550 IFX<=46THENPX=-3*((RM=1)OR(RM=3))-20*
    ((RM=2)OR(RM=4))                :rem  29
```

```
560 IFX>46THENPX=-19*((RM=1)OR(RM=3))-36*
    ((RM=2)OR(RM=4))                 :rem  33
570 PY=-7*((RM=1)OR(RM=2))-16*((RM=3)OR(R
    M=4)):PL=(PY*40)+1024+PX:OPL=PL
                                     :rem 142
580 GOTO330                          :rem 108
590 PX=PX-DX:PY=PY-DY                :rem 135
600 GOTO360                          :rem 104
610 PRINT"{HOME}{ 7 GIU'}{ 16 DES}{RVS}
    {YEL}GAME OVER{OFF}"             :rem  90
611 PRINT"{GIU'}{ 7 DES}{RVS}PREMI UN TAS
    TO PER GIOCARE[<7>]{OFF}":POKE198,0
                                     :rem 118
612 WAIT198,1                        :rem 205
613 PRINT"{CLR}":RETURN              :rem  24
650 POKEOPL,32:POKEOPL+54272,0:POKEPL,81:
    POKEPL+54272,4:GOSUB260:GOSUB290
                                     :rem 250
660 IFPX=18ORPX=35THEN680            :rem  52
670 X=45:Y=145:POKE53264,PEEK(53264)AND25
    4:POKE49189,0:GOTO690            :rem 108
680 X=45:Y=145:POKE53264,PEEK(53264)OR1:P
    OKE49189,1                       :rem 192
690 POKE53248,X:POKE53249,Y:POKE53264,PEE
    K(53264)AND1:POKE53269,0         :rem  50
700 PL=53250:POKEPL,FNRX(58):POKEPL+1,FNR
    Y(67):POKEPL+2,FNRX(169)         :rem 227
710 POKEPL+3,FNRY(67):POKEPL+4,FNRX(58):P
    OKEPL+5,FNRY(146)                :rem  50
720 POKEPL+6,FNRX(167):POKEPL+7,FNRY(146)
                                     :rem 220
730 POKEPL+8,INT(RND(1)*222)+33:POKEPL+9,
    INT(RND(1)*139)+66:POKEEN,255
                                     :rem  64
740 GOSUB900:POKE49178,0             :rem 134
750 SYS49152:A=PEEK(49178)           :rem 219
760 IFA=1THENPOKE49178,0:POKE49190,0:RETU
    RN                               :rem 136
770 IFA<>2THEN800                    :rem 225
780 POKE49178,0:POKE49190,0:P=P-1:IFRM(RM
    )=1THENRM(RM)=0:SC=SC-300:TC=TC-1
                                     :rem 207
785 FORI=20TO0STEP-1:POKELH,I:POKECR,17:F
    ORW=1TO20:NEXT:POKECR,16:NEXT
                                     :rem  82
790 POKECR,0:POKE53261,0:RETURN      :rem 160
800 IFA<>3THEN810                    :rem 221
801 SC=SC+300:GOSUB290:TC=TC+1:POKE49178,
    0:RM(RM)=1:POKE53261,0           :rem 229
802 FORI=0TO100STEP10:POKELH,I:POKECR,17:
    FORW=1TO20:NEXT:POKECR,16:NEXT
                                     :rem 122
803 POKECR,0:GOTO750                 :rem 202
810 POKE49178,0:POKE49190,0:SC=SC+10:GOSU
    B290:GOTO750                     :rem 144
820 POKE2040,230:POKE2041,231:POKE2042,23
    2:POKE2043,233:POKE2044,234      :rem  36
830 POKE2045,236:POKE2046,235        :rem  87
840 CH=0:FORI=14720TO15359:READA:POKEI,A:
    CH=CH+A:NEXT                     :rem  11
845 IFCH<>45651THENPRINT"ERRORE NEI DATI
    SPRITE!!!":END                   :rem  82
850 RETURN                           :rem 125
880 POKEEN,0:PRINT"{CLR}{ 5 DES}
    { 5 GIU'}HAI MIGLIORATO LIVELLO: ";LV
    " !"                             :rem 158
890 FORW=1TO1000:NEXT:RETURN         :rem  68
900 A=PEEK(49189):ONA+1GOTO910,920
                                     :rem 170
910 A=1424:GOTO930                   :rem 242
920 A=1462                           :rem 230
930 FORI=ATOA+120STEP40:POKEI,160:POKEI+1
    ,160:NEXTI:RETURN                :rem 174
960 DATA 0,60,0,0,126,0,0,219        :rem 120
970 DATA 0,0,255,0,0,255,0,0         :rem  70
980 DATA 126,0,0,60,0,1,255,128      :rem 230
990 DATA 3,255,192,3,126,192,3,126
                                     :rem 143
1000 DATA 192,3,126,192,3,126,192,3
                                     :rem 174
1010 DATA 126,192,0,102,0,0,102,0
                                     :rem  43
1020 DATA 0,102,0,0,102,0,0,102      :rem 186
1030 DATA 0,0,102,0,0,231,0,0        :rem  91
1050 DATA 0,0,0,0,0,0,0,0            :rem 148
1060 DATA 112,0,0,216,0,1,240,0      :rem 201
1070 DATA 1,192,0,1,192,0,0,224      :rem 216
1080 DATA 0,0,56,0,0,28,0,0          :rem  12
1090 DATA 14,0,240,14,131,252,14,199
                                     :rem 222
1100 DATA 255,14,207,15,158,254,7,252
                                     :rem  25
1110 DATA 124,3,248,56,0,224,0,0:rem  12
1120 DATA 0,0,0,0,0,0,0,0            :rem 146
1140 DATA 0,0,0,12,0,48,30,0         :rem  54
1150 DATA 120,18,24,72,51,60,204,97
                                     :rem 178
1160 DATA 60,134,13,153,176,30,255,120
                                     :rem  65
1170 DATA 51,127,204,45,255,180,110,255
                                     :rem 121
1180 DATA 118,219,255,219,53,255,172,110
                                     :rem 182
1190 DATA 255,118,219,255,219,48,60,12
                                     :rem  88
1200 DATA 32,24,4,96,0,6,64,0        :rem 127
1210 DATA 2,64,0,2,0,0,0,0           :rem 208
1230 DATA 0,24,0,0,60,0,0,126        :rem 105
1240 DATA 0,0,195,0,0,195,0,0        :rem 115
1250 DATA 102,0,0,60,0,0,255,0       :rem 155
1260 DATA 1,255,128,3,189,192,3,189
                                     :rem 197
1270 DATA 192,7,126,224,7,126,224,9
                                     :rem 189
1280 DATA 126,144,16,255,8,9,126,144
                                     :rem 241
1290 DATA 6,255,96,0,255,0,1,255:rem  36
1300 DATA 128,3,255,192,0,0,0,0      :rem 216
1320 DATA 2,0,64,14,66,112,63,90:rem  23
1330 DATA252,127,255,254,255,129,255,255
                                     :rem 239
1340 DATA 165,255,221,129,187,136,195,17
                                     :rem 192
1350 DATA 0,195,0,0,102,0,0,126      :rem 210
1360 DATA 0,0,60,0,0,60,0,0          :rem   4
1370 DATA 60,0,0,60,0,16,56,0        :rem 119
1380 DATA 48,248,0,255,224,0,127,128
                                     :rem 237
1390 DATA 0,48,0,0,16,0,0,16         :rem  69
1410 DATA 0,0,0,0,0,0,0,0            :rem 148
1420 DATA 0,0,0,0,0,0,0,0            :rem 149
1430 DATA 0,0,0,126,0,1,153,128      :rem 212
1440 DATA 2,60,64,6,255,96,6,255:rem  44
1450 DATA 96,2,60,64,1,153,128,0:rem  30
1460 DATA 126,0,0,0,0,0,0,0          :rem   2
1470 DATA 0,0,0,0,0,0,0,0            :rem 154
1480 DATA 0,0,0,0,0,0,0,255          :rem   7
1490 DATA 0,0,0,0,0,0,0,0            :rem 156
1500 DATA 0,0,0,0,0,0,0,0            :rem 148
1510 DATA 0,0,0,0,0,0,0,0            :rem 149
1520 DATA 0,0,0,63,255,252,96,0      :rem 227
1530 DATA 6,192,0,3,255,255,255,192
                                     :rem 188
```

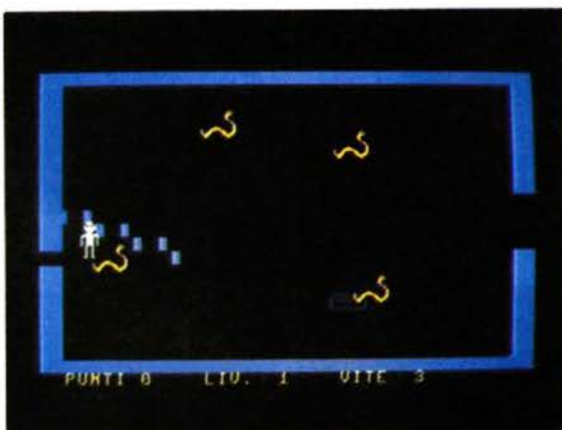
LOAD



Forbidden Crypt

```
1540 DATA 20,3,192,28,3,192,0,3 :rem  229
1550 DATA 192,0,3,255,255,255,0,0
                                :rem   76
1560 DATA 0,0,0,0,0,0,0,255     :rem    6
1570 DATA 0,0,0,0,0,0,0,0       :rem  155
1580 DATA 0,0,0,0,0,0,0,0       :rem  156
1590 DATA 0,0,0,0,0,97,255,134  :rem  177
1600 DATA 146,0,73,147,129,201,137,255
                                :rem   76
1610 DATA 145,135,255,225,128,255,1,65
                                :rem   81
1620 DATA 255,130,34,126,68,28,60,56
                                :rem  240
1630 DATA 0,255,0,3,255,192,31,255
                                :rem  127
1640 DATA 248,0,0,0,0,0,0,255   :rem  115
1650 DATA 1,255,128,3,255,192,7,129
                                :rem  192
1660 DATA 224,12,195,48,24,102,24,56
                                :rem  235
1670 DATA 60,28,60,126,60,54,195,108
                                :rem  242
1680 DATA 51,153,204,49,189,140,49,189
                                :rem  101
1690 DATA 140,51,153,204,54,195,108,60
                                :rem   79
1700 DATA 126,60,56,60,28,24,102,24
                                :rem  175
1710 DATA 12,195,48,7,129,224,3,255
                                :rem  191
1720 DATA 192,1,255,128,0,0,0,0 :rem  220
1730 DATA 0,0,0,0,0,0,0,0       :rem  153
1740 DATA 0,0,0,0,0,126,0,0     :rem    3
1750 DATA 195,0,0,195,0,0,126,0 :rem  226
1760 DATA 0,60,0,0,60,0,0,60    :rem   62
1770 DATA 0,0,126,0,0,255,0,0   :rem  114
1780 DATA 255,0,1,255,128,1,255,128
                                :rem  186
1790 DATA 3,255,192,3,255,192,3,255
                                :rem  196
1800 DATA 192,1,255,128,0,255,0,0
                                :rem   71
1810 CH=0:I=49152:IFPEEK(I)=76THENRETURN
                                :rem  200
1820 READA:CH=CH+A:IFA=256THEN1840
                                :rem    1
1830 POKEI,A:I=I+1:GOTO1820     :rem   84
1840 IFCH<>141279THENPRINT"ERRORI NEI DAT
     I DEL LM!":END             :rem   10
1850 RETURN                     :rem  174
1860 DATA 76,24,194,1,2,4       :rem  205
1870 DATA 8,16,32,64,128,254    :rem  106
1880 DATA 253,251,247,239,223,191
                                :rem  102
1890 DATA 127,0,0,0,0,0         :rem   82
1900 DATA 0,0,0,0,0,0           :rem  224
1910 DATA 0,0,0,0,0,0           :rem  225
1920 DATA 0,0,0,0,0,0           :rem  226
1930 DATA 0,0,173,20,192,10     :rem   29
1940 DATA 168,141,19,192,185,0  :rem  205
1950 DATA 208,141,21,192,172,20 :rem  240
1960 DATA 192,185,3,192,45,16   :rem  159
1970 DATA 208,141,22,192,172,19 :rem  251
1980 DATA 192,185,1,208,141,23  :rem  200
1990 DATA 192,173,21,192,24,109 :rem  255
2000 DATA 24,192,141,21,192,201 :rem  226
2010 DATA 255,240,19,201,0,240  :rem  174
2020 DATA 3,76,129,192,172,20   :rem  142
2030 DATA 192,185,3,192,141,22  :rem  190
2040 DATA 192,76,129,192,173,24 :rem  254
2050 DATA 192,16,12,173,21,192  :rem  189
2060 DATA 201,255,208,5,169,0   :rem  140
2070 DATA 141,22,192,173,23,192 :rem  241
2080 DATA 24,109,25,192,141,23  :rem  190
2090 DATA 192,172,20,192,173,22 :rem  244
2100 DATA 192,240,12,185,3,192  :rem  187
2110 DATA 13,16,208,141,16,208  :rem  182
2120 DATA 76,168,192,185,11,192 :rem  255
2130 DATA 45,16,208,141,16,208  :rem  189
2140 DATA 172,19,192,173,21,192 :rem  247
2150 DATA 153,0,208,173,23,192  :rem  189
2160 DATA 201,66,144,9,201,205  :rem  186
2170 DATA 240,2,176,3,153,1     :rem   35
2180 DATA 208,169,0,141,24,192  :rem  195
2190 DATA 141,25,192,96,173,0   :rem  149
2200 DATA 220,41,15,201,15,208  :rem  172
2210 DATA 3,76,111,193,32,207   :rem  137
2220 DATA 193,32,44,192,173,1   :rem  142
2230 DATA 208,201,66,208,9,24   :rem  143
2240 DATA 105,1,141,1,208,76    :rem   83
2250 DATA 246,192,201,205,208,6 :rem  241
2260 DATA 56,233,1,141,1,208    :rem   85
2270 DATA 173,16,208,41,1,208   :rem  141
2280 DATA 42,173,0,208,201,39   :rem  140
2290 DATA 208,74,173,1,208,201  :rem  193
2300 DATA 130,144,22,201,161,176:rem   21
2310 DATA 18,173,16,208,41,1    :rem   87
2320 DATA 205,37,192,240,8,169  :rem  200
2330 DATA 1,141,26,192,76,111   :rem  137
2340 DATA 193,238,0,208,76,78   :rem  159
2350 DATA 193,173,0,208,201,49  :rem  194
2360 DATA 208,32,173,1,208,201  :rem  185
2370 DATA 130,144,22,201,161,176:rem   28
2380 DATA 18,173,16,208,41,1    :rem   94
2390 DATA 205,37,192,240,8,169  :rem  207
2400 DATA 1,141,26,192,76,111   :rem  135
2410 DATA 193,206,0,208,162,5   :rem  138
2420 DATA 32,54,195,173,32,192  :rem  199
2430 DATA 240,23,169,3,141,26   :rem  139
2440 DATA 192,160,5,185,11,192  :rem  196
2450 DATA 45,21,208,141,21,208  :rem  186
2460 DATA 160,10,169,0,153,0    :rem   83
2470 DATA 208,173,38,192,208,90 :rem    2
2480 DATA 173,43,192,208,85,173 :rem    4
2490 DATA 0,220,41,16,208,78    :rem   94
2500 DATA 173,40,192,201,3,176  :rem  188
2510 DATA 71,169,50,141,43,192  :rem  196
2520 DATA 169,1,141,38,192,238  :rem  202
2530 DATA 40,192,169,10,141,1   :rem  136
2540 DATA 212,169,17,141,4,212  :rem  189
2550 DATA 162,2,32,15,194,169   :rem  150
2560 DATA 16,141,4,212,160,12   :rem  131
2570 DATA 173,0,208,153,0,208   :rem  140
2580 DATA 173,1,208,153,1,208   :rem  143
2590 DATA 173,16,208,41,1,240   :rem  142
2600 DATA 9,173,16,208,9,64     :rem   54
2610 DATA 141,16,208,96,173,16  :rem  199
2620 DATA 208,41,191,141,16,208 :rem  241
2630 DATA 96,168,41,1,208,8     :rem   55
2640 DATA 169,255,141,25,192,76 :rem    4
2650 DATA 230,193,152,41,2,208  :rem  190
2660 DATA 5,169,1,141,25,192    :rem   99
2670 DATA 152,41,4,208,8,169    :rem  104
2680 DATA 255,141,24,192,76,253 :rem    1
```

```
2690 DATA 193,152,41,8,208,5     :rem  104
2700 DATA 169,1,141,24,192,173   :rem  195
2710 DATA 38,192,208,12,173,24   :rem  197
2720 DATA 192,141,39,192,173,25  :rem  253
2730 DATA 192,141,41,192,96,160  :rem  251
2740 DATA 255,136,208,253,202,208
                                 :rem   91
2750 DATA 248,96,169,0,141,20    :rem  153
2760 DATA 192,32,202,192,44,43   :rem  197
2770 DATA 192,240,3,206,43,192   :rem  198
2780 DATA 173,26,192,208,88,173  :rem   11
2790 DATA 38,192,240,21,32,13    :rem  145
2800 DATA 196,173,38,192,240,13  :rem  252
2810 DATA 173,26,192,208,70,32   :rem  198
2820 DATA 13,196,173,26,192,208  :rem  255
2830 DATA 62,174,42,192,32,13    :rem  148
2840 DATA 194,162,0,173,27,192   :rem  203
2850 DATA 24,105,1,201,5,208     :rem   84
2860 DATA 2,169,1,141,27,192     :rem  100
2870 DATA 141,20,192,168,185,3   :rem  203
2880 DATA 192,45,16,208,208,16   :rem  207
2890 DATA 152,10,168,185,0,208   :rem  201
2900 DATA 208,8,232,224,4,240    :rem  140
2910 DATA 161,76,79,194,32,139   :rem  215
2920 DATA 194,173,26,192,208,3   :rem  205
2930 DATA 76,24,194,169,0,141    :rem  155
2940 DATA 38,192,96,173,27,192   :rem  219
2950 DATA 10,168,185,0,208,141   :rem  196
2960 DATA 21,192,185,1,208,141   :rem  197
2970 DATA 23,192,173,23,192,205  :rem  252
2980 DATA 1,208,208,8,169,0      :rem   54
2990 DATA 141,23,192,76,187,194  :rem   12
3000 DATA 144,8,169,255,141,25   :rem  195
3010 DATA 192,76,187,194,169,1   :rem  210
3020 DATA 141,25,192,172,27,192  :rem  243
3030 DATA 185,3,192,45,16,208    :rem  147
3040 DATA 208,8,169,0,141,22     :rem   89
3050 DATA 192,76,211,194,169,1   :rem  202
3060 DATA 141,22,192,173,16,208  :rem  241
3070 DATA 41,1,205,22,192,240    :rem  131
3080 DATA 5,176,18,76,236,194    :rem  163
3090 DATA 173,21,192,205,0,208   :rem  189
3100 DATA 240,12,144,5,169,255   :rem  189
3110 DATA 76,248,194,169,1,76    :rem  162
3120 DATA 248,194,169,0,141,24   :rem  196
3130 DATA 192,173,24,192,73,255  :rem  252
3140 DATA 24,105,1,141,34,192    :rem  132
3150 DATA 173,25,192,73,255,24   :rem  201
3160 DATA 105,1,141,35,192,32    :rem  134
3170 DATA 44,192,32,33,195,173   :rem  202
3180 DATA 32,192,208,1,96,32     :rem   98
3190 DATA 229,195,96,162,0,236   :rem  210
3200 DATA 20,192,240,8,32,54     :rem   85
3210 DATA 195,173,32,192,208,5   :rem  198
3220 DATA 232,224,6,208,238,96   :rem  200
3230 DATA 173,20,192,10,168,185  :rem  245
3240 DATA 0,208,141,28,192,185   :rem  195
3250 DATA 1,208,141,29,192,138   :rem  196
3260 DATA 10,168,185,0,208,141   :rem  191
3270 DATA 30,192,185,1,208,141   :rem  192
3280 DATA 31,192,173,29,192,56   :rem  208
3290 DATA 233,19,205,31,192,176  :rem  251
3300 DATA 11,173,29,192,24,105   :rem  188
3310 DATA 19,205,31,192,176,6    :rem  146
3320 DATA 169,0,141,32,192,96    :rem  148
3330 DATA 172,20,192,185,3,192   :rem  196
3340 DATA 45,16,208,240,2,169    :rem  146
3350 DATA 1,141,33,192,138,168   :rem  197
3360 DATA 185,3,192,45,16,208    :rem  153
3370 DATA 240,2,169,1,205,33     :rem   89
3380 DATA 192,240,46,176,25,173  :rem    0
3390 DATA 28,192,56,233,15,176   :rem  210
3400 DATA 11,205,30,192,176,6    :rem  137
3410 DATA 169,1,141,32,192,96    :rem  149
3420 DATA 169,0,141,32,192,96    :rem  149
3430 DATA 173,28,192,24,105,15   :rem  195
3440 DATA 144,242,205,30,192,144:rem   33
3450 DATA 237,169,1,141,32,192   :rem  198
3460 DATA 96,173,28,192,56,233   :rem  214
3470 DATA 15,176,2,169,0,205     :rem   97
3480 DATA 30,192,176,216,173,28  :rem    0
3490 DATA 192,24,105,15,144,2    :rem  143
3500 DATA 169,255,205,30,192,144:rem   41
3510 DATA 201,169,1,141,32,192   :rem  186
3520 DATA 96,138,240,31,173,34   :rem  199
3530 DATA 192,141,24,192,173,35  :rem  248
3540 DATA 192,141,25,192,32,44   :rem  196
3550 DATA 192,173,34,192,141,24  :rem  249
3560 DATA 192,173,35,192,141,25  :rem  252
3570 DATA 192,32,44,192,96,169   :rem  217
3580 DATA 2,141,26,192,96,160    :rem  152
3590 DATA 12,185,1,208,201,66    :rem  146
3600 DATA 240,106,201,205,240,102
                                 :rem   63
3610 DATA 173,16,208,41,1,208    :rem  140
3620 DATA 7,185,0,208,201,39     :rem   95
3630 DATA 240,88,185,0,208,201   :rem  195
3640 DATA 49,240,81,173,39,192   :rem  210
3650 DATA 141,24,192,173,41,192  :rem  248
3660 DATA 141,25,192,169,6,141   :rem  202
3670 DATA 20,192,32,44,192,160   :rem  196
3680 DATA 12,185,1,208,201,66    :rem  146
3690 DATA 240,52,201,205,240,48  :rem  241
3700 DATA 173,16,208,41,64,208   :rem  197
3710 DATA 10,185,0,208,201,39    :rem  137
3720 DATA 240,34,76,104,196,185  :rem  255
3730 DATA 0,208,201,49,240,24    :rem  137
3740 DATA 162,1,32,54,195,173    :rem  150
3750 DATA 32,192,208,22,232,224  :rem  243
3760 DATA 5,208,243,76,122,196   :rem  208
3770 DATA 169,0,141,26,192,96    :rem  160
3780 DATA 169,0,141,26,192,76    :rem  159
3790 DATA 171,196,138,24,105,3   :rem  205
3800 DATA 141,26,192,138,168,185:rem   52
3810 DATA 11,192,45,21,208,141   :rem  188
3820 DATA 21,208,185,11,192,45   :rem  197
3830 DATA 16,208,141,16,208,152  :rem  244
3840 DATA 10,168,169,0,153,0     :rem   94
3850 DATA 208,169,0,141,38,192   :rem  205
3860 DATA 160,12,153,1,208,96,256
                                 :rem   93
```



Forbidden Crypt
versione per C16

```
20 POKE55,0:POKE56,60:CLR:COLOR0,1,0:COLO
   R4,1,0:COLOR1,7,4
30 DIMMN(4,6),DR(8),RM(4)
40 FORI=1TO4:FORJ=1TO6:READMN(I,J):NEXTJ:
   NEXTI:FORI=1TO8:READDR(I):NEXT
50 FORI=1TO4:RM(I)=1:NEXT
60 DATA0,77,78,40,79,72,0,82,83,1,84,69,0
   ,85,86,1,87,34,0,88,89,40,90,52
70 DATA-40,-39,1,41,40,39,-1,-41
80 DEFFNY(X)=INT(X/40):DEFFNX(X)=X-(FNY(X
   )*40)
90 PR=30
100 GOSUB1280:GOSUB1380
110 VOL5
120 T$="{RVS}{ 40 SPAZI}{OFF}"
130 S$="{RVS}{ 2 SPAZI}{OFF}{ 36 SPAZI}
    {RVS}{ 2 SPAZI}{OFF}":P=5
140 DEFFNA(X)=INT(RND(1)*X)+1
150 DEFFNRX(X)=INT(RND(1)*100)+X:DEFFNRY(
    Y)=INT(RND(1)+58)+Y
160 PL=3572:PX=20:PY=12:OPL=PL
170 LV=1:SC=0:GOSUB330:END
180 PRINT"{CLR}";:PRINT:PRINTT$;S$;S$;:GO
    SUB200:PRINTS$;S$;:GOSUB200:PRINTS$;S
    $;T$
190 PRINT"{HOME}":RETURN
200 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    [<A>]{ 13 *}[<S>]{ 2 SPAZI}[<A>]
    { 13 *}[<S>]{ 2 SPAZI}{RVS}
    { 2 SPAZI}{OFF}";
210 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
220 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
230 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    I{ 13 SPAZI}I{ 2 SPAZI}I{ 13 SPAZI}I
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
240 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
250 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    -{ 13 SPAZI}-{ 2 SPAZI}-{ 13 SPAZI}-
    { 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}";
260 PRINT"{RVS}{ 2 SPAZI}{OFF}{ 2 SPAZI}
    [<Z>]{ 13 *}[<X>]{ 2 SPAZI}[<Z>]
    { 13 *}[<X>]{ 2 SPAZI}{RVS}
    { 2 SPAZI}{OFF}";
270 RETURN
280 PRINT"{CLR}";:PRINT:PRINTT$;
290 FORI=1TO20:PRINTS$;:NEXT:PRINTT$:PRIN
    T"{HOME}":PRINT"{ 9 GIU'}";
300 D$="[<L>][<J>]{ 36 SPAZI}[<L>][<J>]":
    PRINTD$;D$;D$;D$:RETURN
310 PRINT"{HOME}{ 23 GIU'}{YEL}
    { 2 SPAZI}PUNTI"SC"{ 3 SPAZI}LIV. "LV
    "{ 3 SPAZI}VITE "P"[<7>]"
320 PRINT"{HOME}":RETURN
330 REM FUORI STANZA
340 IFTC<>4THEN360
350 TC=0:FORI=1TO4:RM(I)=1:NEXT:LV=LV+1:G
    OSUB1030:IFLV/4=INT(LV/4)THENP=P+1
360 GOSUB180:GOSUB310:IFP=0THEN600
370 IFRM(1)=0THENPOKE3363,160
380 IFRM(2)=0THENPOKE3380,160
390 IFRM(3)=0THENPOKE3723,160
400 IFRM(4)=0THENPOKE3740,160
410 POKEPL,81:POKEPL-1024,4
420 D=DR(JOY(1)AND15)
430 IFD=0THEN420
440 PL=PL+D:IFPEEK(PL)<>32THEN470
450 SOUND1,200,1
460 POKEOPL-1024,0:POKEOPL,32:POKEPL-1024
    ,4:POKEPL,81:OPL=PL:GOTO420
470 IFPEEK(PL)<>9THENPL=OPL:GOTO420
480 PY=FNY(PL-3072):PX=FNX(PL-3072)
490 IFPY=7THENRM=1
500 IFPY=16THENRM=3
510 IFPX>19THENRM=RM+1
520 IFRM=1THENCL=6
530 IFRM=2THENCL=7
540 IFRM=3THENCL=5
550 IFRM=4THENCL=2
560 IFRM(RM)=0THENPL=PL-D:GOTO460
570 PX=FNX(PL-3072):PY=FNY(PL-3072):GOSUB
    640
580 REM
590 GOTO340
600 PRINT"{HOME}{ 7 GIU'}{ 16 DES}{RVS}
    {YEL}GAME OVER{OFF}"
610 PRINT"{GIU'}{ 7 DES}{FLASH ON}PREMI U
    N TASTO PER GIOCARE![<7>][<7>]":POKE2
    39,0

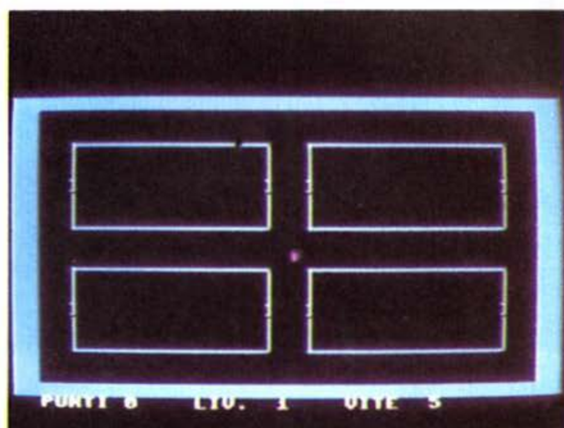
620 WAIT239,1
630 PRINT"{CLR}":RUN
640 GOSUB280:GOSUB310:IFPX=21ORPX=4THENPX
    =2:PY=11:GOTO660
650 PX=37:PY=11
660 GOSUB1050
670 PL=PX+3072+(PY*40):POKEPL,0:POKEPL-10
    24,2:OPL=PL
680 MT=MN(RM,2):MB=MN(RM,3):MP=MN(RM,4):M
    D=MN(RM,5):MC=MN(RM,6)
690 FORI=0TO3:X=FNA(35)+2:Y=FNA(18)+2:Z=3
    072+X+(Y*40):POKEZ+MP,MB:POKEZ,MT
700 POKEZ-1024,MC:POKEZ+MP-1024,MC:M(I)=Z
    :NEXTI
710 X=FNA(18)+1:Y=FNA(12)+1:Z=3072+X+(40*
    Y):IFPEEK(Z)<>32THEN710
720 TR=2+MT:TL=Z:POKETL,TR
730 FL=0:DOWHILEFL=0
740 D=DR(JOY(1)AND15):A=PEEK(PL+D)
750 IFD<>0THENDM=D
760 IFJOY(1)AND128THENGOSUB1130
770 IFA=32THEN810
780 IF(A=0)OR(A=160)THEN830
790 IFA=TRTHENSC=SC+300:RM(RM)=0:TC=TC+1:
    GOSUB310:GOSUB1270:GOTO810
800 FL=1:GOTO840
810 PL=PL+D:POKEOPL-1024,0:POKEOPL,32:POK

    EPL-1024,2:POKEPL,0:OPL=PL
820 SOUND1,100,1
830 TQ=MQ:GOSUB910:IFTQ<>MQTHENFL=1:A=0:F
    ORI=200TO0STEP-5:SOUND1,I,2:NEXT
840 LOOP
850 IFA>100THEN870
860 P=P-1:IFRM(RM)=0THENRM(RM)=1:SC=SC-30
    0:TC=TC-1
870 PX=FNX(PL-3072):IFRM>2THENPY=16:ELSEP
    Y=7
880 IFPX<20THENPX=-20*((RM=2)OR(RM=4))-3*
    ((RM=1)OR(RM=3)):GOTO900
890 PX=-19*((RM=1)OR(RM=3))-36*((RM=2)OR(
    RM=4))
900 PL=3072+PX+(PY*40):OPL=PL:RETURN
910 FORI=1TO4:MM=MM+1:IFMM=4THENMM=0
920 IFM(MM)<>0THENI=6
930 NEXTI
940 IFI=5THENRETURN
```



Forbidden Crypt

```
950 V1=PL-3072:Y1=FNY(V1):X1=FNX(V1):V=M(
    MM)-3072
960 Y=FNY(V):X=FNX(V):M=40*(Y>Y1)-40*(Y<Y
    1):M=M-1*(X<X1)+1*(X>X1):TM=M(MM)+M
970 Q=PEEK(TM):R=PEEK(TM+MP):IFQ=0ORQ=C1O
    RR=C1THENMQ=MQ+1:RETURN
980 IF(M<>MP)AND(M<>-MP)THEN990
981 IFM=MPTHENIF(R<>0)AND(R<>32)THENRETUR
    N
982 IFM=-MPTHENIF(Q<>0)AND(Q<>32)THENRETU
    RN
983 GOTO1000
990 IF(Q<>32)OR(R<>32)THENRETURN
1000 POKEM(MM),32:POKEM(MM)+MP,32:POKETM-
     1024,MC:POKETM+MP-1024,MC:POKETM,MT
1010 POKETM+MP,MB:M(MM)=TM
1020 RETURN
1030 PRINT"{CLR}{ 10 GIU'}{ 6 DES}TU HAI
     SORPASSATO IL LIVELLO NUMERO"LV"!"
1040 FORW=1TO1000:NEXTW:RETURN
1050 IFPX=37THENGOTO1070
1060 A=3472:GOTO1080
1070 A=3510
1080 FORI=ATOA+120STEP40:POKEI,160:POKEI+
     1,160:NEXTI:RETURN
1090 D=JOY(1)
1100 IFD=0THENRETURN
1110 D=DR(D):RETURN
1120 REM FUOCO
1130 MS=PL+DM:OMS=MS:PT=PL
1140 SD=900
1150 B=PEEK(MS)
1160 IF(B=160)OR(B=117)OR(B=118)OR(B=TR)T
     HEN1250
1170 PT=PT+1
1180 IFB=MBORB=MTTHEN1210
1190 SD=SD-10:SOUND1,SD,10
1200 POKEOMS-1024,0:POKEMS-1024,27:POKEMS
     ,PR:POKEOMS,32:OMS=MS:MS=MS+DM:GOTO1
     150
1210 SC=SC+10:GOSUB310
1220 FORI=0TO3
1230 IFM(I)=MSORM(I)+MP=MSTHENPOKEM(I),32
     :POKEM(I)+MP,32:M(I)=0
1240 NEXT
1250 IFPT<>PLTHENPOKEOMS,32:POKEOMS-1024,
     0
1260 RETURN
1270 FORI=100TO1000STEP50:SOUND1,I,2:NEXT
     :RETURN
1280 FORI=819TO869:READA:POKEI,A:NEXT
1290 SYS819
1300 POKE65298,PEEK(65298)AND251
1310 POKE65299,(PEEK(65299)AND3)OR(60)
1320 RETURN
1330 DATA169,0,133,55,169,60,133,56,160
1340 DATA0,185,0,208,153,0,60,200,152
1350 DATA208,246,185,0,209,153,0,61,200,1
     52,208,246,185,0,210,153,0
1360 DATA62,200,152,208,246,185,0,211,153
     ,0,63,200,152,208
1370 DATA246,96
1380 FORI=1TO14:READA:FORJ=15360+(A*8)TO1
     5360+(A*8)+7:READB:POKEJ,B:NEXT:NEXT
1390 RETURN
1400 DATA0,60,126,219,255,189,195,126,60
1410 DATA77,126,66,36,24,126,153,24,126
1420 DATA78,153,24,126,153,24,126,153,0
1430 DATA82,0,0,0,131,198,108,56,0
1440 DATA83,4,24,24,140,204,108,56,0
1450 DATA85,28,54,98,64,113,63,5,5
1460 DATA86,0,0,0,16,248,252,80,80
1470 DATA88,24,36,36,24,126,189,189,189
1480 DATA89,189,60,60,126,255,0,0,0
1490 DATA30,0,0,24,60,60,24,0,0
1500 DATA79,0,0,0,126,153,153,129,255
1510 DATA84,192,224,118,60,28,62,38,0
1520 DATA87,126,231,195,153,153,195,231,1
     26
1530 DATA90,60,24,24,24,60,126,126,60
```



Forbidden Crypt
versione per VIC 20

programma 1

```
110 POKE44,32:POKE642,32:POKE8192,0:POKE6
    48,30:POKE36866,150:POKE36869,240:PRI
    NT"{CLR}"                      :rem  148
130 S$=CHR$(131)                   :rem   47
150 FORI=1TOLEN(S$):POKE630+I,ASC(MID$(S$
    ,I)):NEXT:POKE198,I:END        :rem  141
```

programma 2

```
100 CS=4096:PRINT"{CLR}":FORA=0TO255:POKE
    7680+A,A:POKE38400+A,0:NEXTA:POKE3686
    9,252                          :rem   34
110 FORA=0TO2047:POKEA+CS,PEEK(32768+A):N
    EXTA                           :rem  212
120 FORA=0TO1023:READB:POKECS+A,B:NEXTA:P
    OKE36869,240                   :rem  247
130 FORA=0TO85:READB:POKE7168+A,B:NEXTA
                                   :rem  164
140 REM CARICAMENTO PROGRAMMA SUCCESSIVO
                                   :rem   76
150 S$=CHR$(131)                   :rem   49
170 FORI=1TOLEN(S$):POKE630+I,ASC(MID$(S$
    ,I)):NEXT:POKE198,I:END        :rem  143
```



Forbidden Crypt

```
180 DATA 60,126,219,255,255,219,102,60,0,
    0,0,0,24,60,90,24              :rem  10
190 DATA 0,64,32,240,240,32,64,0,24,90,60
    ,24,0,0,0,0                    :rem 205
200 DATA 0,2,4,15,15,4,2,0,0,56,56,184,14
    4,254,18,18                    :rem 225
210 DATA 16,40,68,68,68,68,198,0,2,63,150
    ,254,18,40,198,0               :rem   1
220 DATA 0,3,118,124,112,0,0,0,0,192,96,5
    1,30,0,0,0                     :rem 142
230 DATA 40,56,214,243,219,14,0,0,60,122,
    247,129,126,122,122,122        :rem  47
240 DATA 30,62,24,24,60,126,126,60,24,52,
    82,255,74,44,24,0              :rem  21
250 DATA 144,96,112,184,28,14,7,3,28,116,
    31,127,15,4,4,12               :rem 232
260 DATA 12,24,240,224,224,96,32,96,20,29
    ,106,252,252,36,36,108         :rem  25
270 DATA 124,66,66,124,72,68,66,0,16,16,1
    6,8,15,0,0,0                   :rem  35
280 DATA 8,8,8,16,240,0,0,0,72,72,84,35,2
    0,8,7,0                        :rem  36
290 DATA 18,18,42,196,40,16,224,0,129,129
    ,66,36,153,189,126,60          :rem 251
300 DATA 60,126,189,153,36,66,129,129,34,
    34,34,220,124,220,34,34        :rem  74
310 DATA 60,90,90,126,36,36,24,0,60,32,32
    ,32,32,32,60,0                 :rem 112
320 DATA 12,16,16,60,16,112,110,0,60,4,4,
    4,4,4,60,0                     :rem 151
330 DATA 0,8,28,42,8,8,8,8,0,0,16,32,127,
    32,16,0                        :rem  30
340 DATA 0,0,0,0,0,0,0,0,8,8,8,8,0,0,8,0
                                   :rem 109
350 DATA 36,36,36,0,0,0,0,0,36,36,126,36,
    126,36,36,0                    :rem 224
360 DATA 8,30,40,28,10,60,8,0,0,98,100,8,
    16,38,70,0                     :rem 178
370 DATA 48,72,72,48,74,68,58,0,60,24,24,
    24,24,24,24,60                 :rem 152
380 DATA 0,0,63,63,63,63,60,60,0,0,252,25
    2,252,252,60,60                :rem 169
390 DATA 60,60,252,252,252,252,0,0,60,60,
    63,63,63,63,0,0                :rem 170
400 DATA 0,0,255,255,255,255,0,0,60,60,60
    ,60,60,60,60,60                :rem 162
410 DATA 0,0,0,0,0,24,24,0,0,2,4,8,16,32,
    64,0                           :rem  99
420 DATA 60,66,70,90,98,66,60,0,8,24,40,8
    ,8,8,62,0                      :rem 155
430 DATA 60,66,2,12,48,64,126,0,60,66,2,2
    8,2,66,60,0                    :rem 237
440 DATA 4,12,20,36,126,4,4,0,126,64,120,
    4,2,68,56,0                    :rem 222
450 DATA 28,32,64,124,66,66,60,0,126,66,4
    ,8,16,16,16,0                  :rem  91
460 DATA 60,66,66,60,66,66,60,0,60,66,66,
    62,2,4,56,0                    :rem   1
470 DATA 0,0,8,0,0,8,0,0,0,0,8,0,0,8,8,16
                                   :rem 168
480 DATA 14,24,48,96,48,24,14,0,0,0,126,0
    ,126,0,0,0                     :rem 169
490 DATA 112,24,12,6,12,24,112,0,60,66,2,
    12,16,0,16,0                   :rem   0
500 DATA 0,64,32,240,240,32,64,0,0,2,4,15
    ,15,4,2,0                      :rem 101
510 DATA 0,0,0,0,24,60,90,24,0,0,0,0,240,
    48,80,144                      :rem  98
520 DATA 0,0,0,0,15,12,10,9,24,90,60,24,0
    ,0,0,0                         :rem 195
530 DATA 144,80,48,240,0,0,0,0,9,10,12,15
    ,0,0,0,0                       :rem  44
540 DATA 4,4,4,4,4,4,4,4,0,0,0,0,224,16,8
    ,8                             :rem  22
550 DATA 8,8,8,4,3,0,0,0,8,8,8,16,224,0,0
    ,0                             :rem  30
560 DATA 128,128,128,128,128,128,128,255,
    128,64,32,16,8,4,2,1           :rem 194
570 DATA 1,2,4,8,16,32,64,128,255,128,128
    ,128,128,128,128,128           :rem 195
580 DATA 255,1,1,1,1,1,1,1,0,60,126,126,1
    26,126,60,0                    :rem 206
590 DATA 0,0,0,0,0,0,255,0,54,127,127,127
    ,62,28,8,0                     :rem 169
600 DATA 64,64,64,64,64,64,64,64,0,0,0,0,
    3,4,8,8                        :rem  43
610 DATA 129,66,36,24,24,36,66,129,0,60,6
    6,66,66,66,60,0                :rem 207
620 DATA 8,28,42,119,42,8,8,0,2,2,2,2,2,2
    ,2,2                           :rem 127
630 DATA 8,28,62,127,62,28,8,0,8,8,8,8,25
    5,8,8,8                        :rem  73
640 DATA 160,80,160,80,160,80,160,80,8,8,
    8,8,8,8,8,8                    :rem   4
650 DATA 0,0,1,62,84,20,20,0,255,127,63,3
    1,15,7,3,1                     :rem 166
660 DATA 0,0,0,0,0,0,0,0,240,240,240,240,
    240,240,240,240                :rem 122
670 DATA 0,0,0,0,255,255,255,255,255,0,0,
    0,0,0,0,0                      :rem 103
680 DATA 0,0,0,0,0,0,0,255,128,128,128,12
    8,128,128,128,128              :rem  16
690 DATA 170,85,170,85,170,85,170,85,1,1,
    1,1,1,1,1,1                    :rem 233
700 DATA 0,0,0,0,170,85,170,85,255,254,25
    2,248,240,224,192,128          :rem 226
710 DATA 3,3,3,3,3,3,3,3,8,8,8,8,15,8,8,8
                                   :rem 204
720 DATA 0,0,0,0,15,15,15,15,8,8,8,8,15,0
    ,0,0                           :rem 117
730 DATA 0,0,0,0,248,8,8,8,0,0,0,0,0,0,25
    5,255                          :rem 166
740 DATA 0,0,0,0,15,8,8,8,8,8,8,8,255,0,0
    ,0                             :rem  35
750 DATA 0,0,0,0,255,8,8,8,8,8,8,8,248,8,
    8,8                            :rem 116
760 DATA 192,192,192,192,192,192,192,192,
    224,224,224,224,224,224,224,224
                                   :rem 235
770 DATA 7,7,7,7,7,7,7,7,255,255,0,0,0,0,
    0,0                            :rem  92
780 DATA 255,255,255,0,0,0,0,0,0,0,0,0,0,
    255,255,255                    :rem 213
790 DATA 1,1,1,1,1,1,1,255,0,0,0,0,240,24
    0,240,240                      :rem  89
800 DATA 15,15,15,15,0,0,0,0,8,8,8,8,248,
    0,0,0                          :rem 172
810 DATA 240,240,240,240,0,0,0,0,240,240,
    240,240,15,15,15,15            :rem  79
820 DATA 169,0,141,52,3            :rem 100
830 DATA 141,56,3,169,127          :rem 211
840 DATA 141,34,145,173,32         :rem 253
850 DATA 145,41,128,208,8          :rem 211
860 DATA 169,1,141,52,3            :rem 105
```

```
870 DATA 76,40,28,173,17          :rem 167
880 DATA 145,41,16,208,5          :rem 159
890 DATA 169,2,141,52,3           :rem 109
900 DATA 173,17,145,41,4          :rem 153
910 DATA 208,12,173,52,3          :rem 150
920 DATA 24,105,3,141,52          :rem 145
930 DATA 3,76,75,28,173           :rem 119
940 DATA 17,145,41,8,208          :rem 160
950 DATA 9,173,52,3,24            :rem  60
960 DATA 105,6,141,52,3           :rem 101
970 DATA 173,17,145,41,32         :rem 209
980 DATA 208,3,238,56,3,96        :rem  14
```

programma 3

```
20 DIMMN(4,6),DR(8),RM(4)              :rem  89
30 FORI=1TO4:FORJ=1TO6:READMN(I,J):NEXTJ:
   NEXTI:FORI=1TO8:READDR(I):NEXT
                                      :rem 194
40 FORI=1TO4:RM(I)=1:NEXT             :rem  51
50 DATA7792,5,6,22,7,1,7802,8,9,1,10,5,79
   46,15,16,1,17,2,7956,23,24,22,25,4
                                      :rem 163
60 DATA1,-1,-22,-21,-23,22,23,21:rem  34
70 DEFFNY(X)=INT(X/22):DEFFNX(X)=X-(FNY(X
   )*22):DEFFNA(X)=INT(RND(1)*X)+1
                                      :rem  73
80 TP$="(,,,,,,,,,,,,,,,,,,,,)":BT$="+,,,
   ,,,,,,,,,,,,,,,,,*"                :rem 172
90 SI$="-{ 20 SPAZI}-":ML$="{RED}@@@@@@@
   @":LV=1:SC=0:MQ=1                  :rem 165
100 POKE36878,10:POKE36879,8:POKE36869,25
    2:C=30720:P=7910                  :rem 199
110 IFTC=4THENTC=0:GOSUB790           :rem 182
120 GOSUB180                          :rem 172
130 GOSUB430:P=(MN(RM,1)+8)+9*(P1=-1)
                                      :rem   9
140 IFMQ<>11THENGOTO110               :rem 152
150 PRINT"{CLR}{ 10 GIU'}{YEL}{ 6 DES}
    {RVS}GAME OVER{OFF}"              :rem 234
160 PRINT"{GIU'}{ 4 DES}{RVS}PREMI UN TAS
    TO":PRINT"{ 4 DES}{RVS}PER GIOCARE
    {OFF}"                            :rem 174
170 POKE198,0:WAIT 198,1:PRINT"{BLU}":RUN
                                      :rem 140
180 GOSUB690:GOSUB710:PRINT"{BLU}":FORI=1
    TO4:IFRM(I)=0THENPOKEMN(I,1)+3,160
                                      :rem  99
190 NEXT:POKEP,46:POKEP+C,2:GOSUB760
                                      :rem   4
200 SYS7168:P1=DR(PEEK(820)):TP=PEEK(P+P1
    )                                 :rem 227
210 IF(TP<>32)AND(TP<>39)THEN200:rem 101
220 P=P+P1:IFTP=32THENPOKEP-P1,32:POKEP+C
    ,2:POKEP,46:GOTO200               :rem 242
230 TP=P:FORI=1TO4:IF(MN(I,1)=P)OR(MN(I,1
    )+7=P)THENRM=I:I=5                :rem 106
240 NEXTI:P=7897+(17*(P=MN(RM,1))):MT=MN(
    RM,2):MB=MN(RM,3):MP=MN(RM,4)
                                      :rem 136
250 MD=MN(RM,5):MC=MN(RM,6)           :rem  69
260 IFRM(RM)=1THENPOKEP-P1,32:POKEP+C,2:P
    OKEP,46:RETURN                    :rem 255
270 P=TP-P1:GOTO200                   :rem 125
280 FORI=1TO4:MM=MM+1:IFMM=4THENMM=0
                                      :rem  32
290 IFM(MM)<>0THENI=6                 :rem 247
300 NEXTI                             :rem  27
310 IFI=5THENRETURN                   :rem 237
320 V1=P-7680:Y1=FNY(V1):X1=FNX(V1):V=M(M
    M)-7680:Y=FNY(V):X=FNX(V)         :rem   2
330 M=22*(Y>Y1)-22*(Y<Y1):M=M-1*(X<X1)+1*
    (X>X1):TM=M(MM)+M:Q=PEEK(TM):rem 239
340 R=PEEK(TM+MP)                     :rem   5
350 IF(Q=0)OR(R=0)THENMQ=MQ+1:GOTO410
                                      :rem 243
360 IF(M<>MP)AND(M<>-MP)THEN400 :rem  85
370 IFM=MPTHENIF(R<>0)AND(R<>32)THENRETUR
    N                                 :rem 191
380 IFM=-MPTHENIF(Q<>0)AND(R<>32)THENRETU
    RN                                :rem 236
390 GOTO410                           :rem 106
400 IF(Q<>32)OR(R<>32)THENRETURN:rem 214
410 POKEM(MM),32:POKEM(MM)+MP,32:POKETM+C
    ,MC:POKETM+MP+C,MC:POKETM,MT:rem 142
420 POKETM+MP,MB:M(MM)=TM:RETURN:rem  83
430 GOSUB690:GOSUB740:GOSUB760:POKEP,0:PO
    KEP+C,2                           :rem 252
440 FORI=0TO3                         :rem  11
441 X=FNA(19):Y=FNA(13):Z=7680+X+(Y*22)
                                      :rem 164
442 IF PEEK(Z)<>32THEN441             :rem 160
443 POKEZ,MT:POKEZ+MP,MB              :rem  55
450 POKEZ+C,MC:POKEZ+MP+C,MC:M(I)=Z:NEXTI
                                      :rem 123
470 X=FNA(18)+1:Y=FNA(12)+1:Z=X+7680+(Y*2
    2):IFPEEK(Z)<>32THEN470           :rem 158
480 TR=10+RM:TL=Z:POKETL,TR:POKETL+C,6
                                      :rem 207
490 SYS7168:P1=PEEK(820):C1=P1:P1=DR(P1)
                                      :rem  70
500 IFPEEK(P+P1+P1)=93THENRETURN:rem 250
510 T=MQ:GOSUB280:TP=PEEK(P+P1+P1):IFMQ=T
    THEN540                           :rem  18
520 IFRM(RM)=0THENRM(RM)=1:SC=SC-300:TC=T
    C-1                               :rem 180
530 RETURN                            :rem 120
540 IF(TP<=45)AND(TP>=40)THENP1=0:C1=0
                                      :rem 222
550 IFPEEK(824)THENGOSUB610:GOTO490
                                      :rem 147
560 IF(P+P1)<>TLAND(P+P1+P1)<>TLTHEN570
                                      :rem  66
561 RM(RM)=0:TC=TC+1:SC=SC+300:GOSUB760:P
    OKETL+C,0:POKETL,32:TL=0:GOTO490
                                      :rem  21
570 G=PEEK(P+P1):H=PEEK(P+P1+P1):IF(G<>MT
    )AND(G<>MB)AND(H<>MT)AND(H<>MB)THEN58
    0                                 :rem  39
575 MQ=MQ+1:IFRM(RM)=0THENRM(RM)=1:TC=TC-
    1:SC=SC-300                       :rem 205
576 RETURN                            :rem 130
580 IFC1=0THEN490                     :rem 217
590 POKEP+P2,32:POKEP,32:P=P+P1:POKEP+C,2
    :POKEP+P1+C,3                     :rem  31
600 POKEP,0:POKEP+P1,63+C1:P2=P1:C2=C1:GO
    TO490                             :rem 244
610 AL=P+P2:S=0:IFC2=0THENP2=22:C2=1:POKE
    36876,200                         :rem 167
620 S=S+1:POKEAL+((S-1)*P2),32:POKE36876,
    200-S*5:BL=PEEK(AL+(S*P2))        :rem  31
630 IF(BL=MT)OR(BL=MB)THENSC=SC+10:GOSUB7
    60:POKE36876,0:GOTO660            :rem 142
640 IFPEEK(AL+(S*P2))<>32THENPOKEP+P2,63+
    C2:POKE36876,0:RETURN             :rem   3
650 POKEAL+(S*P2),63+C2:POKEAL+(S*P2)+C,C
    L+1:GOTO620                       :rem   8
660 AC=AL+(S*P2):FORA=0TO3:POKE36877,200
                                      :rem  64
670 IFAC=M(A)ORAC=M(A)+MPTHENPOKEM(A),32:
    POKEM(A)+MP,32:M(A)=0:POKE36877,0:RET
    URN                               :rem 184
680 NEXTA:POKE36877,0:RETURN          :rem  12
```

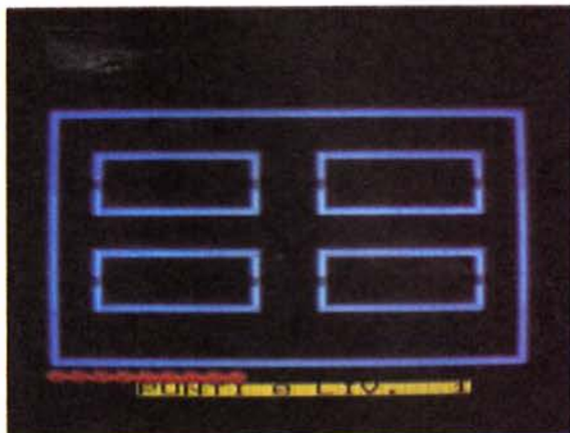
LOAD



Forbidden Crypt

```
690 PRINT"{CLR}";:PRINTTP$;:FORI=1TO17:PR
    INTSI$;:NEXT:PRINTBT$;           :rem 226
700 RETURN                           :rem 119
710 PRINT"{HOME}{ 3 GIU'}";:FORB=1TO2:PRI
    NT"{ 2 DES}(,,,,,,){ 2 SPAZI}(,,,,,,)
    { 2 DES}";                       :rem 252
720 PRINT"{ 2 DES}-{ 6 SPAZI}-{ 2 SPAZI}
    -{ 6 SPAZI}-{ 2 DES}";:PRINT"
    { 2 DES}'{ 6 SPAZI}'{ 2 SPAZI}'
    { 6 SPAZI}'{ 2 DES}";            :rem 35
730 PRINT"{ 2 DES}-{ 6 SPAZI}-{ 2 SPAZI}
    -{ 6 SPAZI}-{ 2 DES}";:PRINT"
    { 2 DES}+,,,,,,*{ 2 SPAZI}+,,,,,,*
    { 2 GIU'}{ 2 DES}";:NEXTB:RETURN
                                     :rem 57
740 PRINT"{HOME}{ 8 GIU'}";          :rem 66
750 FORI=1TO3:PRINT"-{ 20 DES}_";:NEXT:RE
    TURN                             :rem 231
760 PRINT"{HOME}{ 19 GIU'}";         :rem 255
770 PRINTLEFT$(ML$,11-MQ)CHR$(13);:PRINT"
    {RVS}{YEL}{ 4 DES}PUNTI";SC;"{SIN} LI
    V. "LV"{OFF}";                   :rem 74
780 PRINT"{HOME}";:PRINT"{BLU}";:RETURN
                                     :rem 61
790 FORI=1TO4:RM(I)=1:NEXT:LV=LV+1
                                     :rem 134
800 PRINT"{CLR}{ 7 GIU'}{ 7 DES}{RVS}HAI
    RAGGIUNTO":PRINT"{ 3 DES}{RVS}IL LIVE
    LLO"LV"{OFF}"                    :rem 106
810 FORI=1TO2000:NEXT:RETURN         :rem 47

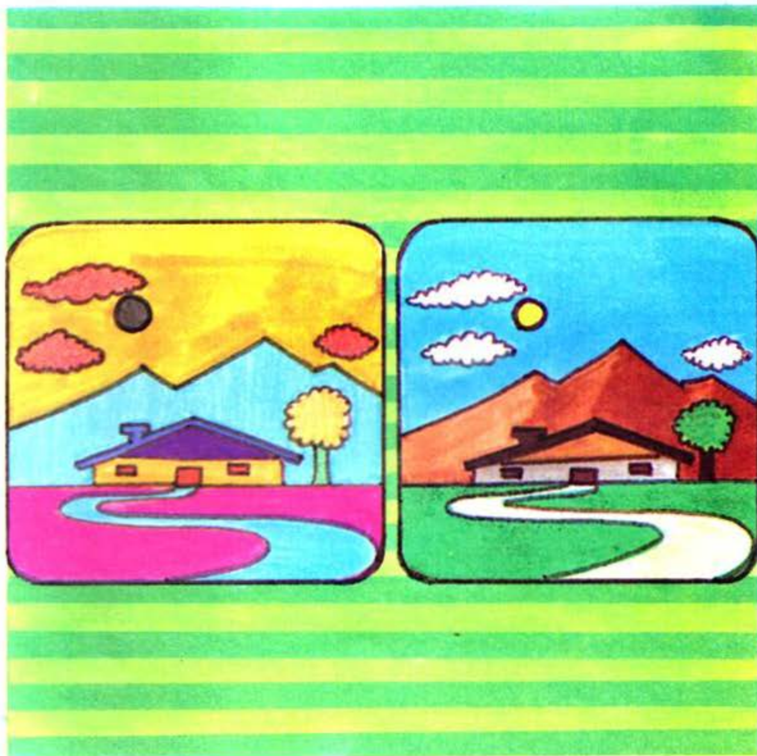
READY.                               :rem 173
```



Color Swap

di **L. Noel Jr.**
trad. ed adatt.
di **M. Cristuib Grizzi**

Siete ormai frustrati dalla perdita di tempo – e dalla complessità – insita nel cambiamento dei colori visualizzati nelle schermate in alta risoluzione?
Questo programma permette di cambiare i colori sullo schermo in modo istantaneo, e funziona inoltre in tutti i diversi modi grafici.



COLOR SWAP è una utility in linguaggio macchina impiegabile dai vostri programmi in BASIC. Funzionando ad alta velocità, vi permetterà di modificare a piacere i colori visualizzati sullo schermo, senza dover listare o comunque "editare" il vostro programma, e senza dover attendere la "ricostruzione" dello schermo da parte del computer.
È inoltre utilizzabile con qualsiasi tipo di visualizzazione: testo, multicolore, alta risoluzione, sprite e così via... COLOR SWAP si regola automaticamente sulla configurazione di schermo che avete scelto precedentemente.
Dopo avere caricato il programma ci sarà una breve attesa, necessaria per inserire in memoria il codice macchina, quindi verrà visualizzato un messaggio contenente i parametri necessari per le chiamate a questa routine.
A questo punto digitate NEW e premete RETURN: la routine in linguaggio macchina risiederà comunque nella memoria del computer, fino a che spegnerete la macchina od inserirete altri valori nelle locazioni di memoria occupate dalla routine stessa.
COLOR SWAP è molto semplice da usare: esso crea due nuovi pseudoregistri di colore nelle locazioni 700 e 701: pensate la locazione 700 come il registro del vecchio colore e la locazione 701 come il registro del nuovo colore.
Se fate due POKE in queste locazioni con due diversi codici colore, e quindi operate la SYS alla routine di COLOR SWAP, ogni colore visualizzato che corrisponde al codice inserito nella locazione 700 verrà modificato nel colore il cui codice è contenuto nella locazione 701.
Per rendere più chiaro il concetto, ecco un esempio.
Dopo aver caricato e mandato in esecuzione COLOR SWAP date il NEW e caricate un qualunque programma in BASIC con il quale volete fare esperimenti. Lavorando in modo diretto, potrete ad esempio digitare:

POKE700,6:POKE701,7:SYS XXXXX
(dove XXXXX è l'indirizzo visualizzato da COLOR SWAP)

Premendo RETURN, farete in modo che la routine in linguaggio macchina cerchi tutto ciò che è visualizzato in blu (codice 6) e lo trasformi in giallo (codice 7). Ciò potrà, ad esempio, modificare un intero cielo blu in giallo, o provocare la stessa trasformazione in un piccolo carattere ridefinito.
La modifica dei colori avviene in modo istantaneo e, se non vi piace il risultato, invertite i valori delle POKE e rifate la SYS: ritornerete nelle condizioni iniziali.
Man mano che prenderete confidenza con l'utilizzo di COLOR SWAP vi accorgerete che ci sono alcune cose importanti da tenere a mente: il programma è stato progettato per modificare qualsiasi visualizzazione di un particolare co-

lore, e ciò in alcuni casi può essere sconcertante. Se modificate un cielo blu, ed il colore del testo è anch'esso blu, entrambi verranno sostituiti nel colore prescelto. Inoltre, può risultare difficile usare COLOR SWAP in modo multicolore nel VIC 20 ed in modo colore di fondo esteso con il C64. Il programma funziona bene, ma questi modi grafici, per la loro complessità di utilizzo, richiedono che il programmatore sappia sempre esattamente che cosa sta facendo. Infine, per sicurezza, usate nelle due POKE solo i numeri corrispondenti ai codici di colore disponibili sul computer che state utilizzando.
Notate inoltre che difficilmente COLOR SWAP funzionerà correttamente in congiunzione con altre routine in linguaggio macchina, comprese alcune cartridge come la Super Expander ed altre estensioni del BASIC.

Come è fatto il programma

I vari blocchi di codice macchina lavorano come le subroutine in BASIC, e COLOR SWAP è semplicemente un insieme di subroutine in codice macchina combinate insieme nella giusta sequenza.
La struttura del programma può essere immaginata come una serie di villaggi collegati da una strada a senso unico: se volete, ad esempio, evitare di passare dal villaggio che cambia il colore del testo, immettetevi sulla strada dopo il villaggio. Con COLOR SWAP, una volta entrati nella strada a senso unico dovete necessariamente percorrerla fino alla fine, dove il controllo verrà ripassato al BASIC.
I diagrammi riportati mostrano come è strutturato il programma: le prime sezioni rappresentano modifiche che possono essere effettuate dal BASIC, generalmente con una POKE in una particolare locazione di memoria.
Potete evitare di "passare" da queste porzioni iniziali di programma, evitando così di ottenere alcuni cambiamenti di colore non desiderati.
Evitare di "passare" da un certo blocco di programma è molto semplice: basta fare una SYS all'indirizzo del primo "villaggio", lungo la strada, dal quale volete "passare": ciò, naturalmente, dopo aver effettuato le due POKE con i codici di colore desiderati.
Questa caratteristica di molteplici punti di ingresso posseduta da COLOR SWAP è ovviamente utilissima, e con un minimo di esperienza e sperimentazione potrete ottenere effetti sorprendenti.

Rilocazione del programma

COLOR SWAP è rilocabile. Se la zona di memoria in cui esso risiede "per default" vi serve per altri scopi, potete tranquillamente trasferirlo in qualunque altra zona di memoria libera.
La linea 120 del programma caricatore in BASIC contiene la variabile RA, che può essere modificata con qualsiasi valore (congruente) desiderato. In ogni caso, se rilocate COLOR SWAP nell'area normalmente occupata dai programmi BASIC, dovete prendere alcuni accorgimenti, per evitare possibili sovrascritture.
Dal momento che il VIC 20 possiede una memoria molto limitata, all'utente di questa macchina viene fornita l'opzione di locare COLOR SWAP nel buffer registratore (locazioni 828-1019).
I programmi in linguaggio macchina che occupano queste locazioni sono protetti dal BASIC, ma vengono distrutti dalle operazioni con l'unità a cassette. Così, per aiutare i possessori di VIC 20 la versione di COLOR SWAP per questo computer può rilocarsi, sistemandosi automaticamente in cima alla memoria di qualsiasi configurazione disponiate. Per l'utilizzo di questa opzione è sufficiente rispondere alle domande poste dal computer.
Gli utilizzatori del VIC 20 che ridefiniscono la variabile RA possono cancellare le linee da 10 a 60 ed inserire il valore prescelto nell'espressione RA=-RA all'inizio della linea 120.
Si ricordi che, con qualsiasi rilocazione, gli indirizzi a cui vengono fatte le SYS devono essere congruentemente modificati. L'espressione tra parentesi nel diagramma mostra come ottenere i nuovi indirizzi (RA è il valore del numero che appare dopo la SYS nel messaggio generato da COLOR SWAP).
COLOR SWAP offre inoltre ai possessori di VIC 20 una notevole particolarità: registri colore separati per bordo e fondo.
Normalmente, infatti, il colore del fondo e del bordo sono nel VIC 20 combinati in un unico numero, contenuto nel registro in 36879; con COLOR SWAP, invece, il colore del fondo è contenuto nella locazione 785, mentre il colore del bordo è contenuto nella locazione 786. Il contenuto di queste locazioni può inoltre essere letto normalmente da qualsiasi programma tramite un'istruzione PEEK.

Avvertenze per l'utilizzo con VIC 20

COLOR SWAP opera in maniera così semplice, che non può essere esteso in modo da coprire il "reverse" sul VIC 20. Tuttavia, usando il programma "Invertitore per VIC 20" si ottiene l'inversione dei bit da un valore all'altro ogni volta che si esegue una SYS743. Proprio come COLOR SWAP, anche questo programma può essere rilocato, modificando il valore della variabile RA. Crediamo tuttavia che raramente si possa presentare in questo caso la necessità di una rilocazione: il programma è totalmente indipendente da COLOR SWAP, sicuro da sovrascritture da parte del BASIC e non risiede nel buffer del registratore.

Esempi di possibili utilizzi

Dopo un po' di sperimentazione con COLOR SWAP potrete trovare una moltitudine di utilizzi diversi.
Benché esso sia stato concepito fondamentalmente per l'utilizzo in modo diretto, in modo da "editare" i colori visualizzati, può tranquillamente essere utilizzato in molte altre maniere.
Ad esempio, i possessori di C64 dovrebbero sapere che non è possibile visualizzare un carattere "pokandolo" direttamene nella memoria di schermo, a meno che venga effettuata anche la corrispondente POKE nella memoria colore.
Con COLOR SWAP, conoscendo i valori contenuti nella memoria colore, è possibile creare un contrasto tra il colore di fondo e quello del testo con una SYS alle routine di COLOR SWAP.
Con COLOR SWAP in memoria nel vostro C64, e lo schermo nei colori standard, provate questa linea in modo diretto:

```
POKE700,6:POKE701,1:
SYS49286:POKE1524,83
```

Vedrete un cuoricino bianco apparire vicino al centro dello schermo. Fate delle POKE di altri codici carattere nella mappa video, e li vedrete apparire direttamente sullo schermo: conveniente, vero?
COLOR SWAP può inoltre essere aggiunto ai programmi come subroutine: come parte di lunghi programmi COLOR SWAP permette di avere finestre di testo che lampeggiano rapidamente in diversi colori, scritte o disegni che compaiono o spariscono, e permette di modificare istantaneamente il colore di caratteri e sprite.
L'unica limitazione è la fantasia e la capacità del programmatore.

Locazioni di COLOR SWAP

VIC 20 – inizio

828	(RA)	cambia il colore testo corrente
844	(RA+16)	cambia il colore ausiliario
875	(RA+47)	cambia il colore di fondo
906	(RA+78)	cambia il colore del bordo
942	(RA+114)	cambia i colori rimanenti

Per verificare il colore di fondo: SYS828:PRINTPEEK(785)
Per verificare il colore del bordo: SYS828:PRINTPEEK(786)

C64 – inizio

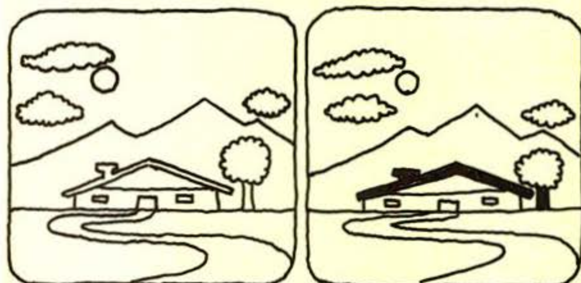
49152	(RA)	cambia il colore testo corrente
49171	(RA+19)	cambia il colore del bordo
49190	(RA+38)	cambia il colore di

	fondo
49196 (RA+44)	cambia il colore degli sprite
49254 (RA+102)	cambia il colore di fondo esteso
49286 (RA+134)	cambia i colori rimanenti

NOTA: Se COLOR SWAP è rilocato, usate le espressioni tra parentesi per calcolare i nuovi indirizzi, in base al nuovo indirizzo RA. Gli utenti di VIC 20 che rilocano automaticamente attraverso il caricatore BASIC **devono** ricalcolare in questo modo gli indirizzi.

Color Swap
versione per C64

```
10 REM *COLOR-SWAP C64*   N E' LM, A E' L
   'INDIRIZZO, E I E' IL CONTATORE.
                                :rem  38
15 PRINT"{CLR}ATTENDERE..."     :rem 240
20 N=74:FORA=685TO690STEP5:FORI=0TO3:POKE
   A+I,N:NEXTI:POKEA+I,96:N=N-64:NEXTA
                                :rem 113
30 CK=0:FORI=0TO9:X=PEEK(685+I):CK=CK+X:N
   EXT                          :rem 252
40 REM CK E' IL CHECKSUM E X UNO DEI SUOI
   COMPONENTI.                  :rem 245
50 IFCK<>528THENPRINT"{CLR}{GIU'}CONTROLL
   A LA LINEA 20!!!":END        :rem 185
100 CK=0:FORA=702TO750:READN:POKEA,N:CK=C
    K+N:NEXTA                   :rem 145
110 IFCK<>5158THENPRINT"{CLR}{GIU'}CONTRO
    LLA LE LINEE DA 702-750!!!":END
                                :rem 167
120 RA=49152:RA$=STR$(RA):L=LEN(RA$):RA$=
    RIGHT$(RA$,L-1):REM RA IS RELOC. ADR.
                                :rem 228
130 CK=0:FORA=RATORA+392:READN:POKEA,N:CK
    =CK+N:NEXTA                 :rem  78
140 IFCK<>46713THENPRINT"{CLR}{GIU'}CONTR
    OLLA LE LINEE DA 49152-49542!!!":END
                                :rem 180
150 PRINT"{CLR}{GIU'}FAI POKE700,OC:POKE7
    01,NC:SYS"RA$:PRINT"PER SCAMBIARE I C
    OLORI,"                     :rem 195
160 PRINT:PRINT"DOVE OC E'IL CODICE DEL V
    ECCHIO COLORE"              :rem  49
170 PRINT"E NC E'IL CODICE DEL NUOVO COLO
    RE.":END                    :rem  86
702 DATA 216,173,188,2,41,15    :rem  97
708 DATA 141,188,2,173,189,2    :rem 109
714 DATA 41,15,141,189,2,169    :rem 103
720 DATA 0,141,172,2,141,171    :rem  80
726 DATA 2,141,169,2,141,170    :rem  93
732 DATA 2,96,173,33,208,41     :rem  51
738 DATA 15,205,188,2,208,6     :rem  56
744 DATA 173,189,2,141,33,208   :rem 154
750 DATA 96                     :rem  37
49152 DATA 32,190,2,173,134,2   :rem 145
49158 DATA 41,15,205,188,2,208  :rem 208
49164 DATA 9,173,189,2,141,134  :rem 212
49170 DATA 2,32,190,2,173,32    :rem  94
49176 DATA 208,41,15,205,188,2  :rem 208
49182 DATA 208,9,173,189,2,141  :rem 214
49188 DATA 32,208,32,190,2,32   :rem 153
49194 DATA 222,2,32,190,2,173   :rem 149
49200 DATA 37,208,41,15,205,188 :rem 252
49206 DATA 2,208,6,173,189,2    :rem 108
49212 DATA 141,37,208,173,38,208:rem  50
49218 DATA 41,15,205,188,2,208  :rem 205
49224 DATA 6,173,189,2,141,38   :rem 161
49230 DATA 208,160,0,185,39,208 :rem 253
49236 DATA 41,15,205,188,2,208  :rem 205
49242 DATA 6,173,189,2,153,39   :rem 165
49248 DATA 208,200,192,8,208,235:rem  53
49254 DATA 32,190,2,173,17,208  :rem 204
49260 DATA 41,64,240,22,32,222  :rem 191
49266 DATA 2,173,36,208,41,15   :rem 156
49272 DATA 205,188,2,208,6,173  :rem 211
49278 DATA 189,2,141,36,208,24  :rem 215
49284 DATA 144,7,173,22,208,41  :rem 207
49290 DATA 16,240,37,169,255,141:rem  55
49296 DATA 170,2,173,34,208,41  :rem 207
49302 DATA 15,205,188,2,208,6   :rem 152
49308 DATA 173,189,2,141,34,208 :rem   4
49314 DATA 173,35,208,41,15,205 :rem 250
49320 DATA 188,2,208,6,173,189  :rem 216
49326 DATA 2,141,35,208,173,17  :rem 203
49332 DATA 208,41,32,208,5,169  :rem 206
49338 DATA 255,141,169,2,173,0  :rem 208
49344 DATA 221,41,3,201,3,208   :rem 139
49350 DATA 4,162,0,134,254,201  :rem 190
49356 DATA 2,208,4,162,64,134   :rem 156
49362 DATA 254,201,1,208,4,162  :rem 196
49368 DATA 128,134,254,201,0,208:rem  47
49374 DATA 4,162,192,134,254,173:rem  56
49380 DATA 24,208,41,240,32,173 :rem 249
49386 DATA 2,32,180,2,101,254   :rem 147
49392 DATA 133,254,169,0,133,253:rem  52
49398 DATA 169,0,133,251,169,216:rem  63
49404 DATA 133,252,162,0,142,167:rem  41
49410 DATA 2,162,0,142,168,2    :rem  91
49416 DATA 162,234,142,183,2,160:rem  44
49422 DATA 0,177,251,41,15,205  :rem 196
49428 DATA 188,2,208,5,173,189  :rem 224
49434 DATA 2,145,251,173,169,2  :rem 207
49440 DATA 201,255,208,2,240,55 :rem 246
49446 DATA 177,253,41,15,141,171:rem  52
49452 DATA 2,205,188,2,208,6    :rem 106
49458 DATA 173,189,2,141,171,2  :rem 212
49464 DATA 177,253,41,240,141,172
                                :rem 101
49470 DATA 2,32,173,2,205,188   :rem 154
49476 DATA 2,208,13,173,189,2   :rem 163
49482 DATA 32,178,2,24,109,171  :rem 209
49488 DATA 2,24,144,7,173,172   :rem 164
```



```
49494 DATA 2,24,109,171,2,145   :rem  155
49500 DATA 253,200,204,168,2,240:rem   33
49506 DATA 2,208,170,174,183,2  :rem  204
49512 DATA 224,96,208,1,96,238  :rem  217
49518 DATA 167,2,230,252,230,254:rem   47
49524 DATA 162,3,236,167,2,240  :rem  203
49530 DATA 2,208,144,162,232,142:rem   39
49536 DATA 168,2,162,96,142,183 :rem   13
49542 DATA 2,208,132            :rem  220
```

Color Swap
versione per VIC 20

```
10 HT=PEEK(644):LT=PEEK(643):PRINT"{CLR}S
   E USI IL NASTRO,{ 5 SPAZI}PREMI {RVS}C
   {OFF}."                       :rem  201
20 MT=HT*256+LT:MT=MT-183:HT=INT(MT/256):
   LT=MT-HT*256:PRINT"{GIU'}ALTRIMENTI PR
   EMI {RVS}D{OFF}."             :rem  156
30 GETR$                         :rem  185
40 IFR$="D"THENRA=828:GOTO80     :rem  170
50 IFR$="C"THENPOKE55,LT:POKE56,HT:CLR:RA
   =PEEK(56)*256+PEEK(55):GOTO80:rem   17
60 GOTO30                        :rem    2
70 REM *COLOR-SWAP VIC-20*  N E'L/M, A E'
   L' INDIRIZZO ED I IL CONTATORE
                                 :rem  157
80 N=74:FORA=680TO685STEP5:FORI=0TO3:POKE
   A+I,N:NEXTI:POKEA+I,96:N=N-64:NEXTA
                                 :rem  118
90 CK=0:FORI=0TO9:X=PEEK(680+I):CK=CK+X:N
   EXT                           :rem  253
100 REM CK E'IL CECKSUM E X UNO DEI SUOI
    COMPONENTI.                  :rem  218
110 IFCK<>528THENPRINT"{CLR}{ 2 SPAZI}
    {GIU'}CONTROLLA LA LINEA 80!!!":END
                                 :rem  236
130 CK=0:FORA=RATORA+182:READN:POKEA,N:CK
    =CK+N:NEXTA                  :rem   75
140 IFCK<>19910THENPRINT"{CLR}{GIU'}CONTR
    OLLA LE LINEE 828-1008!!!":PRINT:END
                                 :rem   83
150 PRINT"{CLR}{GIU'}ORA:":PRINT:PRINT"PO
    KE 700,OC:POKE701,NC":PRINT"SYS"RA:PR
    INT:PRINT                    :rem  102
160 PRINT"DOVE{ 2 SPAZI}OC E'IL VECCHIO C
    ODICE COLORE":PRINT"E NC E' IL NUOVO
    CODICE COLORE"               :rem  121
170 END                          :rem  111
828 DATA 173,134,2,41,15,205     :rem   95
834 DATA 168,2,208,6,173,189     :rem  117
840 DATA 2,141,134,2,173,14      :rem   36
846 DATA 144,41,15,141,19,3      :rem   47
852 DATA 173,14,144,32,168,2     :rem  100
858 DATA 205,188,2,208,12,173    :rem  157
864 DATA 189,2,32,173,2,109      :rem   56
870 DATA 19,3,141,14,144,173     :rem   97
876 DATA 15,144,41,240,141,16    :rem  146
882 DATA 3,32,168,2,141,17       :rem  255
888 DATA 3,205,188,2,208,12      :rem   56
894 DATA 173,189,2,141,17,3      :rem   59
900 DATA 32,173,2,141,16,3       :rem  241
906 DATA 173,15,144,41,8,141     :rem   98
912 DATA 19,3,173,15,144,41      :rem   46
918 DATA 7,141,18,3,205,188      :rem   57
924 DATA 2,208,6,173,189,2       :rem    6
930 DATA 141,18,3,13,19,3        :rem  197
936 DATA 13,16,3,141,15,144      :rem   43
942 DATA 169,0,168,170,133,251   :rem  203
948 DATA 173,2,144,41,128,208    :rem  155
954 DATA 6,169,148,133,252,208   :rem  213
960 DATA 4,169,150,133,252,177   :rem  206
```

LOAD

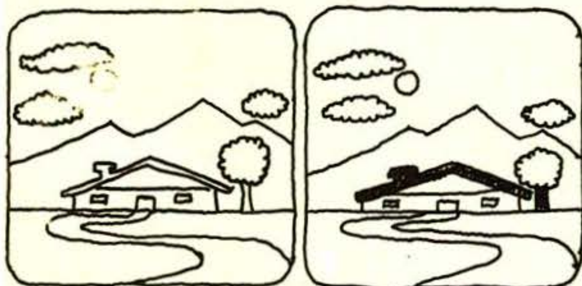


Color Swap

```
966 DATA 251,41,15,205,188,2     :rem  104
972 DATA 208,5,173,189,2,145     :rem  112
978 DATA 251,200,208,239,230,252:rem    46
984 DATA 232,224,1,208,232,177   :rem  203
990 DATA 251,41,15,205,188,2     :rem  101
996 DATA 208,5,173,189,2,145     :rem  118
1002 DATA 251,200,192,251,240,2  :rem  223
1008 DATA 208,235,96             :rem  222
```

Invertitore per VIC 20

```
10 REM *VIC INVERTER* CK =  CHECKSUM,A =
   INDIRIZZO, N = L/M.           :rem   33
20 RA=743:RA$=STR$(RA):L=LEN(RA$):RA$=RIG
   HT$(RA$,L-1):REM RA=IND.RILOCAZIONE
                                 :rem  141
30 CK=0:FORA=RATORA+24:READN:POKEA+I,N:CK
   =CK+N:NEXTA                   :rem   89
40 IFCK<>2391THENPRINT"{CLR}{GIU'}"SPC(3)
   "CONTROLLA LE DATA!!!":END    :rem  233
50 PRINT"{CLR}{GIU'}SYS"RA$" PER ATTIVARE
    E DISATTIVARE IL REVERSE.":END
                                 :rem  144
743 DATA 173,15,144,41,8,240     :rem   97
749 DATA 9,173,15,144,41,247     :rem  111
755 DATA 141,15,144,96,173,15    :rem  156
761 DATA 144,9,8,141,15,144      :rem   51
767 DATA 96                      :rem   45
```

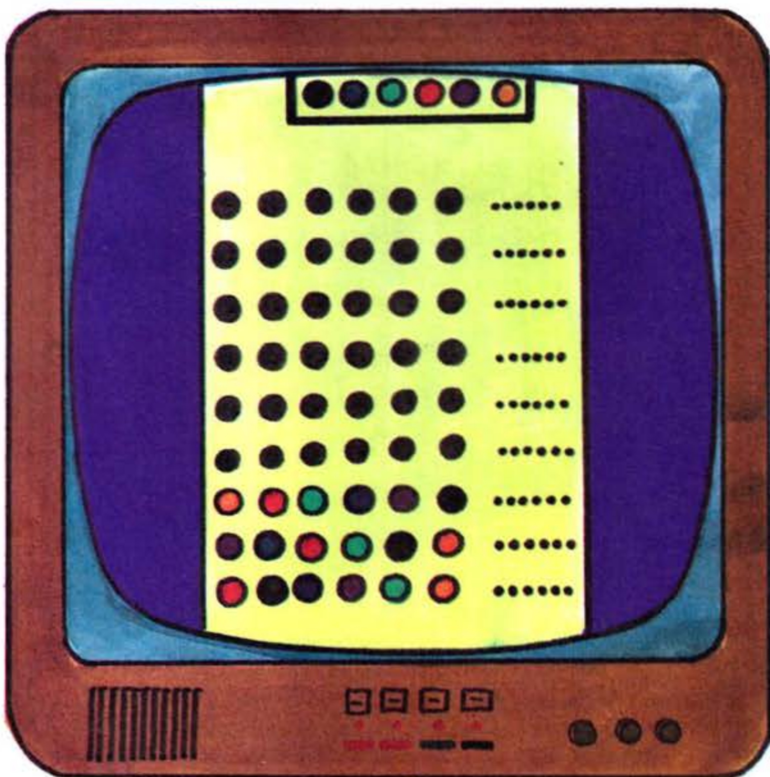




Cypher

di **J.L. Calvin**
trad. e adatt.
di **M. Cristuib Grizzi**

In questo gioco basato sul ragionamento deduttivo la logica ed un po' di fortuna vi aiuteranno nell'indovinare il codice segreto



Questo gioco è basato sull'idea del popolare "Master mind". Dopo aver caricato il programma ed averlo attivato con il RUN vi verrà chiesto se desiderate leggere le istruzioni. All'inizio del gioco appaiono sei diamanti rossi nell'angolo superiore sinistro dello schermo: sotto questi diamanti è nascosto un codice di colori composto da sei elementi. Scopo del gioco è raffigurare e copiare questo codice nel minor numero di tentativi possibile.
Potete inserire la vostra scelta premendo i tasti corrispondenti ai colori desiderati (i numeri da 1 a 8 nella parte superiore sinistra della tastiera).
Come ciascun tasto viene premuto il colore corrispondente alla scelta viene visualizzato sullo schermo, accompagnato da una nota musicale (la sequenza di tutti gli otto colori disponibili suona una scala musicale). Dopo che sono stati premuti sei tasti il computer confronterà le vostre scelte con il codice nascosto sotto i diamanti. Il punteggio viene visualizzato a destra della vostra scelta.
Il codice segreto può contenere più volte lo stesso colore: ad esempio, può contenere due volte il blu o tre volte il bianco.

Punteggio

Esistono tre simboli diversi per indicare il punteggio raggiunto in ciascun round:
- il segno "+" indica che uno dei colori scelti è corretto e si trova nella giusta posizione (quest'ultima non corrisponde necessariamente alla locazione del "+" nell'indicatore di punteggio);
- uno "0" indica che uno dei colori è corretto, ma la sua posizione è sbagliata;
- una "@" indica che il colore è errato, cioè non appartiene al codice segreto.

Ad esempio, supponiamo che il vostro punteggio appaia in questo modo:

++00 @ @

Ciò significa che due delle vostre scelte rappresentano i colori corretti nel posto corretto, due delle vostre scelte rappresentano i colori corretti nella posizione sbagliata e due delle vostre scelte rappresentano colori che non esistono nel codice segreto.
Tenete sempre presente che la posizione dell'indicatore di punteggio non corrisponde alla posizione del colore relativo all'interno del codice segreto.

Disponete di 12 round per indovinare la corretta combinazione di colori: man mano che procedete studiate attentamente i punteggi delle scelte precedenti, al fine di dedurre logicamente quale sia il codice segreto. Se premete il tasto sbagliato, potete usare il tasto DEL per annullare l'intera scelta, e se "date forfait" e desiderate vedere quale sia il codice segreto, premete il tasto "9": il codice verrà visualizzato, e naturalmente perderete la partita.
Un punteggio di nove o dieci tentativi è molto buono, di sei, sette o otto è eccezionale, e di cinque o meno è... un po' troppo fortunato!

Cypher
versione per C64

```
70 PRINT"{CLR}":POKE53280,6:POKE53281,12
                                    :rem 147
80 GOSUB1210                        :rem 172
90 PRINT:PRINT"{CLR}{WHT}{RVS}ISTRUZIONI?
   { 2 SPAZI}S/N{OFF}"              :rem 108
100 GETA$:IFA$=""THEN100            :rem  69
110 IFA$="S"THENGOTO1060            :rem 137
120 IFA$<>"N"THEN100                :rem  83
130 PRINT"{CLR}":DIMA(5),Z(5),S(5),RC(8),
    GC(8)                           :rem 109
140 GLOC=1039:CGLOC=55311:ANS=1024:A$="":
    POKE54296,15                    :rem 231
150 HF=54273:LF=54272:HP=54275:LP=54274:A
    D=54277:SR=54278:WF=54276       :rem 233
160 CANS=55296:SLOC=1053:X=0        :rem 116
170 POKE649,1                       :rem 199
180 FORK=1TO12                      :rem  63
190 FORI=1TO6                       :rem  17
200 POKEGLOC,81:POKECGLOC,11        :rem 218
210 GLOC=GLOC+2:CGLOC=CGLOC+2       :rem  27
220 NEXTI                           :rem  28
230 GLOC=GLOC+68:CGLOC=CGLOC+68     :rem 149
240 NEXTK                           :rem  32
250 GLOC=1039:CGLOC=55311           :rem 164
260 FORL=0TO5                       :rem  16
270 B=INT(RND(1)*8)                 :rem  28
280 RC(B)=RC(B)+1                   :rem 131
290 A(L)=B                          :rem 248
300 POKEANS,90:POKECANS,2:GOSUB1510:ANS=A
    NS+2:CANS=CANS+2                :rem  92
310 NEXTL                           :rem  31
320 ANS=1024:CANS=55296             :rem  34
330 FORI=0TO5                       :rem  11
340 GETZ$:IFZ$=""THEN340            :rem 131
350 IFASC(Z$)=20THENGOTO1020        :rem 151
360 IFASC(Z$)<49ORASC(Z$)>57THEN340
                                    :rem  46
370 Z(I)=VAL(Z$)                    :rem 125
380 Z(I)=Z(I)-1                     :rem  30
390 IFZ(I)=8THEN800                 :rem  91
400 GC(Z(I))=GC(Z(I))+1             :rem 203
410 POKECGLOC,Z(I)                  :rem  76
420 GOSUB1330                       :rem 221
430 CGLOC=CGLOC+2                   :rem   1
440 NEXTI                           :rem  32
450 FORI=0TO7                       :rem  16
460 IFGC(I)<=RC(I)THENQ=Q+GC(I)     :rem  82
470 IFGC(I)>RC(I)THENQ=Q+RC(I)      :rem  35
480 NEXTI                           :rem  36
490 FORK=0TO5                       :rem  20
500 IFZ(K)=A(K)THENF=F+1            :rem 136
510 NEXTK                           :rem  32
520 IFQ=0THEN610                    :rem 170
530 IFF=0THEN580                    :rem 166
540 FORK=0TOF-1                     :rem 127
550 S(K)=91                         :rem  48
560 NEXTK                           :rem  37
570 IFF=QTHEN610                    :rem 197
580 FORK=FTOQ-1                     :rem 164
590 S(K)=87                         :rem  57
600 NEXTK                           :rem  32
610 FORK=0TO5                       :rem  14
620 POKESLOC,S(K)                   :rem  19
630 POKESLOC+54272,0                :rem 132
640 SLOC=SLOC+2                     :rem 150
650 NEXTK                           :rem  37
660 SLOC=SLOC+68                    :rem 212
670 CGLOC=CGLOC+68                  :rem  67
680 I=0                             :rem  84
690 X=X+1                           :rem 232
700 IFF=6THEN800                    :rem 166
710 IFX=12GOTO800                   :rem 240
720 FORK=0TO5                       :rem  16
730 S(K)=0                          :rem 246
740 NEXTK                           :rem  37
750 Q=0:F=0                         :rem  71
760 FORK=0TO7                       :rem  22
770 GC(K)=0                         :rem  49
780 NEXTK                           :rem  41
790 GOTO330                         :rem 111
800 FORT=0TO5                       :rem  24
810 POKEANS,81                      :rem  63
820 POKECANS,A(T)                   :rem   0
830 GOSUB1420                       :rem 226
840 ANS=ANS+2:CANS=CANS+2           :rem  24
850 NEXT                            :rem 220
860 IFZ(I)=8OR(X=12ANDF<6)GOTO910
                                    :rem 222
870 PRINT"":PRINT"":PRINT""         :rem 134
880 PRINT"{BLK} HAI VINTO!!":PRINT"{GIU'}
    {BLK} MOSSE:"X                  :rem 106
890 GOTO920                         :rem 117
910 PRINT"{BLK}{ 3 GIU'} HAI PERSO!!"
                                    :rem 203
920 PRINT:PRINT                     :rem 239
930 PRINT"{ 14 GIU'} PREMI UN TASTO"
                                    :rem   6
940 GETC$:IFC$=""THEN940            :rem  97
960 PRINT"{CLR}{WHT}{ 2 GIU'}{ 2 SPAZI}VU
    OI GIOCARE ANCORA?{ 2 SPAZI}{RVS}S/N
    {OFF}"                          :rem 206
970 GETA$                           :rem 229
980 IFA$=""THEN970                  :rem 229
990 IFA$="S"THENRUN90               :rem 247
1000 IFA$<>"N"THEN970               :rem 144
1009 PRINT"{CLR}":POKE54296,0       :rem 253
1010 POKE53280,14:POKE53281,6:PRINT"[<7>]
     ":POKE649,10:END               :rem 161
1020 FORDE=I*2TO1STEP-1:POKECGLOC-DE,12:G
     OSUB1510:NEXTDE                :rem  28
1030 FORRE=0TO8:GC(RE)=0:NEXTRE     :rem 179
1040 CGLOC=CGLOC-(I*2)              :rem 245
1050 GOTO330                        :rem 149
1060 PRINT"{CLR}DEVI TROVARE UN CODICE DI
      COLORI."                      :rem  67
1065 PRINT"PUOI SCEGLIERE 6 TRA GLI 8 COL
     ORI."                          :rem 148
1070 PRINT"UN COLORE PUO' ESSERE SCELTO P
     IU'";                          :rem 252
1075 PRINT"DI UNA VOLTA.QUANDO APPAIONO I
      6 DIAMANTI,"                  :rem 212
1080 PRINT"INSERISCI LE TUE SCELTE CON I
     TASTI DI{ 2 SPAZI}COLORE."     :rem 175
1090 PRINT"DOPO AVER INSERITO I TUOI 6 CO
     LORI,{ 5 SPAZI}VEDRAI IL TUO PUNTEGG
     IO."                           :rem  68
1100 PRINT"":PRINT"IL PUNTEGGIO E':"
                                    :rem 139
```

```
1110 PRINT"+=COLORE GIUSTO IN POSTO GIUST
     O"                            :rem  162
1120 PRINT"0=COLORE GIUSTO IN POSTO ERRAT
     O"                            :rem  154
1130 PRINT"@=COLORE ERRATO":PRINT""
                                   :rem  175
1140 PRINT"LE POSIZIONI DEI SEGNALINI DEL
       PUNTEGGIO"                  :rem  223
1145 PRINT"NON SI RIFERISCONO AL POSTO DE
     I COLORI"                     :rem   34
1150 PRINT"HAI 12 ROUND, SE VUOI VEDERE "
                                   :rem   23
1160 PRINT"IL CODICE ED USCIRE DAL GIOCO
     PREMI 9.":PRINT""             :rem   90
1170 PRINT"SE HAI INSERITO MENO DI 5 COLO
     RI,"                          :rem   86
1175 PRINT"PREMI 'DEL' PER CANCELLARE LA
     SCELTA."                      :rem  103
1180 PRINT"{RVS}{BLK}PREMI UN TASTO{OFF}"
                                   :rem  122
1190 GETR$:IFR$=""THEN1190         :rem  219
1200 GOTO130                       :rem  144
1210 PRINT"{WHT}{CLR}{ 10 GIU'}{ 9 DES}
     {RVS}{ 3 SPAZI}{OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS}{ 3 SPAZI}{OFF}
     {RVS} {OFF} {RVS} {OFF} {RVS}
     { 3 SPAZI}{OFF} {RVS}{ 3 SPAZI}{OFF}
     "                             :rem  252
1220 PRINTSPC(9)"{RVS} {OFF}{ 3 SPAZI}
     {RVS}{ 3 SPAZI}{OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS} {OFF}[<O>]{RVS}
     {OFF} {RVS} {OFF}[< 2 O>] {RVS}
     {OFF} {RVS} {OFF}"            :rem  245
1230 PRINTTAB(9)"{RVS} {OFF}{ 4 SPAZI}
     {RVS} {OFF}{ 2 SPAZI}{RVS}
     { 2 SPAZI}{OFF}{ 2 SPAZI}{RVS} [<O>]
      {OFF} {RVS} [< 2 O>]{OFF} {RVS}
     { 2 SPAZI}[<*>]{OFF}"         :rem  218
1240 FORD=1TO9:PRINT" ";:NEXTD     :rem   63
1250 PRINT"{RVS}{ 3 SPAZI}{OFF}
     { 2 SPAZI}{RVS} {OFF}{ 2 SPAZI}{RVS}
      {OFF}{ 3 SPAZI}{RVS} {OFF} {RVS}
     {OFF} {RVS}{ 3 SPAZI}{OFF} {RVS}
     {OFF} {RVS} {OFF}"            :rem  185
1310 FORD=1TO2000:NEXT             :rem   60
1320 RETURN                        :rem  166
1330 IFZ(I)=0THENHI=34:LO=75:GOTO1520
                                   :rem  212
1340 IFZ(I)=1THENHI=38:LO=126:GOTO1520
                                   :rem    7
1350 IFZ(I)=2THENHI=43:LO=52:GOTO1520
                                   :rem  211
1360 IFZ(I)=3THENHI=45:LO=198:GOTO1520
                                   :rem   18
1370 IFZ(I)=4THENHI=51:LO=97:GOTO1520
                                   :rem  223
1380 IFZ(I)=5THENHI=57:LO=172:GOTO1520
                                   :rem   17
1390 IFZ(I)=6THENHI=64:LO=188:GOTO1520
                                   :rem   24
1400 IFZ(I)=7THENHI=68:LO=149:GOTO1520
                                   :rem   18
1410 RETURN                        :rem  166
1420 IFA(T)=0THENHI=34:LO=75:GOTO1520
                                   :rem  198
1430 IFA(T)=1THENHI=38:LO=126:GOTO1520
                                   :rem  249
1440 IFA(T)=2THENHI=43:LO=52:GOTO1520
                                   :rem  197
1450 IFA(T)=3THENHI=45:LO=198:GOTO1520
                                   :rem    4
1460 IFA(T)=4THENHI=51:LO=97:GOTO1520
                                   :rem  209
1470 IFA(T)=5THENHI=57:LO=172:GOTO1520
                                   :rem    3
1480 IFA(T)=6THENHI=64:LO=188:GOTO1520
                                   :rem   10
1490 IFA(T)=7THENHI=68:LO=149:GOTO1520
                                   :rem   13
1500 RETURN                        :rem  166
1510 HI=45:LO=198                  :rem  178
1520 POKEHF,HI:POKELF,LO           :rem    4
1530 POKEHP,0:POKELP,255           :rem  185
1540 POKEAD,9:POKESR,0             :rem   77
1550 POKEWF,65                     :rem   46
1560 FORSO=1TO150:NEXTSO           :rem   23
1570 POKEWF,64                     :rem   47
1580 RETURN                        :rem  174
```

Cypher
versione per VIC 20

```
90 POKE36879,232:GOSUB1210:PRINT:PRINT"
    {CLR}{BLK}{RVS}{ 3 SPAZI}ISTRUZIONI
    { 2 SPAZI}S/N{ 4 SPAZI}{OFF}":rem  115
100 GETA$:IFA$=""THEN100           :rem   69
110 IFA$="S"THENGOTO1060           :rem  137
120 IFA$<>"N"THEN100               :rem   83
130 PRINT"{CLR}":DIMA%(5),Z%(5),S%(5),RC%
    (8),GC%(8)                     :rem   38
140 GLOC=7688:CGLOC=38408:ANS=7680:A$="":
    POKE36878,15                   :rem   19
150 HF=36876:LF=36875              :rem  133
160 CANS=38400:SLOC=7695:X=0:FORK=1TO12:F
    ORI=1TO6                       :rem   11
210 POKEGLOC,81:POKECGLOC,1:GLOC=GLOC+1:C
    GLOC=CGLOC+1:NEXTI             :rem   44
230 GLOC=GLOC+38:CGLOC=CGLOC+38:NEXTK:GLO
    C=7688:CGLOC=38408             :rem  178
260 FORL=0TO5:B=INT(RND(1)*8):RC%(B)=RC%(
    B)+1:A%(L)=B                   :rem  246
300 POKEANS,90:POKECANS,2:GOSUB1510:ANS=A
    NS+1:CANS=CANS+1:NEXTL         :rem   31
320 ANS=7680:CANS=38400            :rem   36
330 FORI=0TO5                      :rem   11
340 GETZ$:IFZ$=""THEN340           :rem  131
350 IFASC(Z$)=20THENGOTO1020       :rem  151
360 IFASC(Z$)<49ORASC(Z$)>57THEN340
                                   :rem   46
```

Cypher

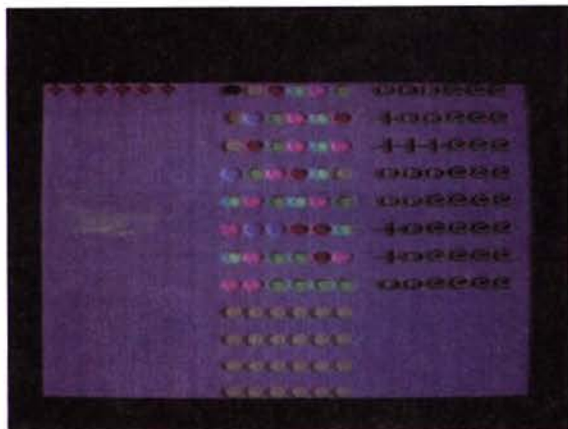
```
370  Z%(I)=VAL(Z$):Z%(I)=Z%(I)-1  :rem  169
390  IFZ%(I)=8THEN800             :rem  128
400  GC%(Z%(I))=GC%(Z%(I))+1      :rem   95
410  POKECGLOC,Z%(I):GOSUB1330:CGLOC=CGLOC
     +1:NEXT:FORI=0TO7            :rem  191
460  IFGC%(I)<=RC%(I)THENQ=Q+GC%(I)
                                  :rem  193
470  IFGC%(I)>RC%(I)THENQ=Q+RC%(I)
                                  :rem  146
480  NEXTI:FORK=0TO5              :rem  213
500  IFZ%(K)=A%(K)THENF=F+1       :rem  210
520  NEXTK:IFQ=0THEN620           :rem  111
530  IFF=0THEN580                 :rem  166
540  FORK=0TOF-1                  :rem  127
550  S%(K)=91:NEXTK               :rem   25
570  IFF=QTHEN620                 :rem  198
580  FORK=FTOQ-1                  :rem  164
590  S%(K)=87:NEXTK               :rem   34
620  FORK=0TO5:POKESLOC,S%(K):POKESLOC+307
     20,0                         :rem    6
640  SLOC=SLOC+1                  :rem  149
650  NEXTK                        :rem   37
660  SLOC=SLOC+38                 :rem  209
670  CGLOC=CGLOC+38               :rem   64
680  I=0                          :rem   84
690  X=X+1                        :rem  232
700  IFF=6THEN800                 :rem  166
710  IFX=12GOTO800                :rem  240
720  FORK=0TO5                    :rem   16
730  S%(K)=0                      :rem   27
740  NEXTK                        :rem   37
750  Q=0:F=0                      :rem   71
760  FORK=0TO7                    :rem   22
770  GC%(K)=0                     :rem   86
780  NEXTK                        :rem   41
790  GOTO330                      :rem  111
800  FORT=0TO5                    :rem   24
810  POKEANS,81                   :rem   63
820  POKECANS,A%(T)               :rem   37
830  GOSUB1420                    :rem  226
840  ANS=ANS+1:CANS=CANS+1        :rem   22
850  NEXTT                        :rem   48
860  POKE198,0:IFZ%(I)=8OR(X=12ANDF<6)GOTO
     910                          :rem  106
880  PRINT"{CLR}{ 3 GIU'}HAI VINTO IN"X"MO
     SSE":PRINT"{ 8 SPAZI}";      :rem  153
890  GOTO930                      :rem  118
910  PRINT"{ 3 GIU'}{BLK}HAI PERSO"
                                  :rem  137
930  PRINT"{ 2 GIU'}{RVS}PREMI{ 2 GIU'}
     { 4 SIN}UN{ 2 GIU'}{ 3 SIN}TASTO"
                                  :rem  219
940  GETC$:IFC$=""THEN940         :rem   97
960  PRINT"{CLR}{ 2 GIU'}{BLK}GIOCHI ANCOR
     A?{RVS}S/N{OFF}"             :rem  207
980  GETA$:IFA$=""THEN980         :rem  101
990  IFA$="S"THENRUN130           :rem   34
1000 IFA$<>"N"THEN980             :rem  145
1010 PRINT"{CLR}":END             :rem   55
1020 FORDE=ITO1STEP-1:POKECGLOC-DE,12:GOS
     UB1510:NEXTDE                :rem  192
1030 FORRE=0TO8:GC%(RE)=0:NEXTRE:rem  216
1040 CGLOC=CGLOC-(I*1)            :rem  244
1050 GOTO330                      :rem  149
1060 PRINT"{CLR}QUESTO E'UN GIOCO DI
     { 2 SPAZI}ABILITA'. 6 COLORI
     { 4 SPAZI}CASUALI SARANNO SCEL-"
                                  :rem  202
1070 PRINT"TI DAL COMPUTER. UN{ 3 SPAZI}
     COLORE PUO' ESSERE{ 4 SPAZI}SCELTO P
     IU' DI UNA"                  :rem   61
1080 PRINT"VOLTA. USA I TASTI DI COLORE P
     ER EFFETTUARE LE TUE SCELTE."
                                  :rem  149
1090 PRINT"DOPO AVER INSERITO I
     { 2 SPAZI}TUOI COLORI VERRA'
     { 4 SPAZI}VISUALIZZATO"      :rem   22
1091 PRINT"IL PUNTEGGIO."         :rem   17
1092 PRINT"{ 5 SPAZI}{GIU'}{RVS}PREMI UN
     TASTO"                       :rem  107
1095 GETA$:IFA$=""THEN1095        :rem  193
1100 PRINT"{CLR}{GIU'}PUNTEGGI:":rem  212
1110 PRINT"{GIU'}+=COLORE GIUSTO
     { 9 SPAZI}POSTO GIUSTO"      :rem   28
1120 PRINT"{GIU'}0=COLORE GIUSTO
     { 9 SPAZI}POSTO SBAGLIATO"   :rem  221
1130 PRINT"{GIU'}@=COLORE SBAGLIATO"
                                  :rem  126
1150 PRINT"{GIU'}HAI 12 TENTATIVI,E SE VU
     OI VEDERE IL"                :rem   50
1160 PRINT"CODICE E FINIRE IL{ 4 SPAZI}GI
     OCO, PREMI '9'."             :rem  166
1170 PRINT"{GIU'}SE HAI SCELTO MENO
     { 4 SPAZI}DI 6 COLORI,PREMI"
                                  :rem   66
1175 PRINT"'DEL' PER CANCELLARE
     { 2 SPAZI}LA SCELTA."        :rem  234
1180 PRINT"{ 5 SPAZI}{RVS}PREMI UN TASTO"
                                  :rem   88
1190 GETR$:IFR$=""THEN1190        :rem  219
1200 GOTO130                      :rem  144
1210 PRINT"{WHT}{CLR}{ 10 GIU'}{ 7 DES}CY
     PHER"                        :rem  109
1310 FORD=1TO2000:NEXTD           :rem  128
1320 RETURN                       :rem  166
1330 IFZ%(I)=0THENHI=130:LO=130:GOTO1520
                                  :rem   78
1340 IFZ%(I)=1THENHI=175:LO=175:GOTO1520
                                  :rem   98
1350 IFZ%(I)=2THENHI=183:LO=183:GOTO1520
                                  :rem   98
1360 IFZ%(I)=3THENHI=191:LO=191:GOTO1520
                                  :rem   98
1370 IFZ%(I)=4THENHI=195:LO=195:GOTO1520
                                  :rem  108
1380 IFZ%(I)=5THENHI=201:LO=201:GOTO1520
                                  :rem   86
1390 IFZ%(I)=6THENHI=207:LO=207:GOTO1520
                                  :rem  100
1400 IFZ%(I)=7THENHI=209:LO=209:GOTO1520
                                  :rem   97
1410 RETURN                       :rem  166
1420 IFA%(T)=0THENHI=130:LO=130:GOTO1520
                                  :rem   64
1430 IFA%(T)=1THENHI=175:LO=175:GOTO1520
                                  :rem   84
1440 IFA%(T)=2THENHI=183:LO=183:GOTO1520
                                  :rem   84
1450 IFA%(T)=3THENHI=191:LO=191:GOTO1520
                                  :rem   84
```



LOAD

Cypher

```
1460  IFA%(T)=4THENHI=195:LO=195:GOTO1520
                                  :rem    94
1470  IFA%(T)=5THENHI=201:LO=201:GOTO1520
                                  :rem    72
1480  IFA%(T)=6THENHI=207:LO=207:GOTO1520
                                  :rem    86
1490  IFA%(T)=7THENHI=209:LO=209:GOTO1520
                                  :rem    92
1500  RETURN                      :rem   166
1510  HI=45:LO=198                :rem   178
1520  POKEHF,HI:POKELF,LO         :rem     4
1560  FORSO=1TO150:NEXTSO         :rem    23
1570  POKEHF,0:POKELF,0           :rem    61
1580  RETURN                      :rem   174
```





Supergraphic per C64

di **G. Kiziak**
trad. e adatt.
di **S. Albarelli**
e **M. Anticoli**

Il C64 è un computer altamente competitivo, a basso costo e con caratteristiche grafiche e sonore molto buone.
Però ha una carenza: non possiede, implementate nella memoria ROM, routine che permettano l'utilizzazione facile e pratica di tali possibilità, che, per essere sfruttate in tutta la loro potenzialità, richiedono tecniche di programmazione complesse.
SUPERGRAPHIC è un insieme di favolose routine, che permette di gestire lo schermo con estrema facilità e velocità.



L'insieme di routine raggruppate in questo utilissimo programma sono state studiate appositamente per manipolare in vari modi il testo e i caratteri presenti sullo schermo, creando effetti speciali abbastanza suggestivi e utili, per rendere un qualsiasi programma più movimentato e divertente da usare.

Il concetto della finestra programmabile

Il normale schermo visualizzato dal C64 in modo testo è formato da una griglia di 25*40 caratteri, che nel programma devono essere indicati con numeri da 0 a 24 per l'altezza e da 0 a 39 per la lunghezza.
Spesso si desidera lavorare solo in una parte dell'intero schermo, che parte in termini informatici è chiamata comunemente "finestra".
Per definire la posizione e le dimensioni della finestra, che deve essere programmata per un dato utilizzo, è necessario tenere conto di quattro parametri principali:

1) la colonna che determina la posizione del lato sinistro della finestra;
2) la riga che determina la posizione del lato superiore della finestra;
3) la larghezza della finestra espressa in colonne;
4) l'altezza della finestra espressa in righe.

In generale, per creare una finestra si possono usare le variabili:
LC..... colonna che determina la posizione del lato sinistro della finestra;
TR..... riga che determina la posizione del lato superiore della finestra;
WIDTH.. larghezza della finestra espressa in colonne;
DEPTH.. altezza della finestra espressa in righe.

Negli esempi esplicativi che seguiranno saranno utilizzate queste variabili nelle istruzioni.
È necessario notare che i parametri LC e TR vengono utilizzati allo stesso modo delle coordinate su un piano cartesiano.

Perciò, se si volesse, ad esempio, creare una finestra grande quanto la schermata, si devono settare i parametri in questo sistema:
LC = 0 TR = 0 WIDTH = 40
DEPTH = 25

I nuovi comandi utilizzabili

I comandi disponibili utilizzando SUPERGRAPHIC si possono dividere in cinque sezioni principali:

1) comandi per lo scroll;
2) comandi per il reverse;
3) comandi per il colore;
4) comandi per il "fill";
5) comandi per disegnare cornici.

Prima di utilizzare le routine in linguaggio macchina è necessario definire, per un loro utilizzo più veloce e pratico, alcune variabili di questo tipo:

UP	=	12 * 4096 + 256
DWN	=	UP + 67
LFT	=	UP + 295
RHT	=	UP + 369
RVS	=	UP + 447
COL	=	UP + 511
FILL	=	UP + 547
BRDR	=	UP + 611

Queste variabili conterranno l'indirizzo di partenza della routine di cui recano le iniziali o l'abbreviazione secondo questo criterio:

UP	=	Routine di scroll verso l'alto
DWN	=	Routine di scroll verso il basso
LFT	=	Routine di scroll verso sinistra
RHT	=	Routine di scroll verso destra
RVS	=	Routine di reverse
COL	=	Routine di colorazione
FILL	=	Routine di riempimento
BRDR	=	Routine di formazione cornici.

I comandi di scroll

Ci sono quattro diverse routine di scroll ognuna delle quali può assumere due variazioni per un totale di otto tipi di scroll diversi!!

1) SCROLL SU
Sintassi: SYS UP, LC, TR, WIDTH, DEPTH, W

2) SCROLL GIÙ
Sintassi: SYS DWN, LC, TR, WIDTH, DEPHT, W

3) SCROLL A SINISTRA
Sintassi: SYS LFT, LC, TR, WIDTH, DEPHT, W

4) SCROLL A DESTRA
Sintassi: SYS RHT, LC, TR, WIDHT, DEPTH, W

Effetto: (per tutti e quattro i tipi di routine) dopo aver definito i quattro parametri per la grandezza e la posizione della finestra sullo schermo è necessario definire la variabile W, chiamata variabile di ritorno.
Se essa vale 0, i caratteri che usciranno dalla finestra in uno qualsiasi dei lati verranno cancellati.
Se invece W è diversa da 0, i caratteri che usciranno dalla finestra torneranno in essa, entrando dalla parte opposta al lato dal quale erano usciti.

I comandi di scroll possono essere utili in diverse circostanze e per diversi scopi.
Ad esempio, è possibile creare presentazioni in movimento o cancellare una schermata in modo elegante, molto originale e pratico.

I comandi di reverse

Esistono due tipi di comandi in questa sezione dell'utility SUPERGRAPHIC:

1) REVERSE/MODO 1
Sintassi: SYS RVS, LC, TR, WIDTH, DEPTH
Effetto: questo modo dei comandi per gestire finestre in reverse inverte tutti i caratteri presenti nella finestra, lasciando i colori come erano prima dell'inversione.

2) REVERSE/MODO 2
Sintassi: SYS RVS, LC, TR, WIDTH, DEPTH, CR
Effetto: questo modo dei comandi per gestire finestre in reverse inverte tutti i caratteri presenti nella finestra, inserendo il colore definito dalla variabile CR in tutta la finestra ed ignorando quello che i caratteri possedevano prima (la variabile CR può assumere un valore da 0 a 15).
I comandi di reverse possono servire per far lampeggiare tutto ciò che si trova in una data finestra o per creare effetti speciali.

Il comando di colorazione

Sintassi: SYS COL, LC, TR, WIDTH, DEPTH, CR
Effetto: questo comando può cambiare il colore di tutti i caratteri presenti in una data finestra mediante la variabile CR, che contiene il numero corrispondente al colore che devono assumere tali caratteri (la variabile CR può assumere un valore da 0 a 15).
Questo comando può servire per far leggermente lampeggiare i caratteri in una finestra o per attrarre l'attenzione su un testo più importante degli altri.

I comandi di riempimento o "FILL"

Esistono due tipi di comandi di riempimento:

1) FILL/MODO 1
Sintassi: SYS FILL, LC, TR, WIDTH, DEPTH, CH
Effetto: riempie una data finestra con il carattere specificato mediante la variabile CH (la variabile CH può assumere un valore da 0 a 255 e corrisponde al valore che si utilizza per pokare un carattere sullo schermo).
Questo comando può servire per cancellare una sezione dello schermo (CH = 32 [spazio]) o riempire una finestra con un motivo decorativo.

2) FILL/MODO 2
Sintassi: SYS FILL, LC, TR, WIDTH, DEPTH, CH, CR
Effetto: riempie una data finestra con il carattere specificato mediante la variabile CH (la variabile CH può assumere un valore da 0 a 255 e corrisponde al valore che si utilizza per pokare un carattere sullo schermo) e lo colora con il colore specificato nella variabile CR (CR può assumere un valore da 0 a 15)

I comandi di formazione cornici

Sintassi 1: SYS BRDR, LC, TR, WIDTH, DEPTH, CH
Sintassi 2: SYS BRDR, LC, TR, WIDTH, DEPTH, CH, CR
Effetti: il primo comando disegna il bordo della finestra, usando un carattere determinato mediante la variabile CH (CH può assumere un valore da 0 a 255).
Il secondo comando genera l'effetto del primo e in più colora il carattere con il quale disegna la cornice con un colore definito dalla variabile CR (CR può assumere un valore da 0 a 15).

Il programma dimostrativo

Per avere una dimostrazione delle possibilità offerte da questo fantastico programma è stato appositamente creato un programma dimostrativo, che permette di capire ancor meglio come funzionano le routine di SUPERGRAPHIC e poterle poi utilizzare in altri programmi.

Note per l'utilizzo

Quando si utilizzano le routine che formano il programma SUPERGRAPHIC è necessario accertarsi che prima siano state definite le variabili UP, DWN, LFT, RHT, RVS, COL, FILL, BRDR, perché in caso contrario il computer si bloccherebbe molto facilmente.
Inoltre, se si utilizzano le routine in un programma di propria invenzione, è necessario accertarsi che non vi siano variabili uguali nel programma.
Prima di far girare il programma dimostrativo, o qualsiasi programma che utilizzi le routine di SUPERGRAPHIC, è necessario caricare in memoria le routine in L.M..



Superghaphic per C64

```
0 FORI=49408TO50121:READX:POKEI,X:NEXT
                                  :rem  159
1 DATA32,226,193,32,241,183,134,2,32,20,1
  94,165,251,166,252,32           :rem   17
2 DATA155,193,32,205,193,166,254,202,48,2
  1,24,165,208,105,40,133         :rem  121
3 DATA210,133,214,165,209,105,0,133,211,3
  2,119,193,202,16,235,164        :rem  151
4 DATA253,185,32,192,145,208,185,72,192,1
  45,212,136,16,243,32,20         :rem  130
5 DATA194,88,96,32,226,193,32,241,183,134
  ,2,32,20,194,32,182             :rem  195
6 DATA194,170,165,251,32,155,193,32,205,1
  93,166,254,202,240,208,56       :rem  235
7 DATA165,208,233,40,133,210,133,214,165,
  209,233,0,133,211,24,32         :rem  100
8 DATA119,193,202,16,234,48,184,41,3,9,21
  6,133,215,164,253,177           :rem   40
9 DATA210,145,208,177,214,145,212,136,16,
  245,165,211,133,209,165,210     :rem   64
10 DATA133,208,165,215,133,213,165,214,13
   3,212,96,72,169,216,133,208    :rem   66
11 DATA172,136,2,136,132,209,24,165,208,1
   05,40,133,208,165,209,105      :rem  213
12 DATA0,133,209,202,16,240,24,104,101,20
   8,133,208,133,212,165,209      :rem  191
13 DATA105,0,133,209,41,3,9,216,133,213,1
   64,253,96,165,2,208            :rem  173
14 DATA3,169,32,44,177,208,153,32,192,177
   ,212,153,72,192,136,16         :rem   89
15 DATA236,96,32,241,183,138,41,63,133,25
   1,32,241,183,138,41,31         :rem   76
16 DATA133,252,32,241,183,202,134,253,24,
   165,251,101,253,201,40,176     :rem  254
17 DATA16,32,241,183,202,134,254,24,165,2
   52,101,254,201,25,176,1        :rem  109
18 DATA96,76,72,178,120,162,30,181,208,72
   ,189,0,192,149,208,104         :rem   99
19 DATA157,0,192,202,16,241,96,32,226,193
   ,32,241,183,134,2,32           :rem  229
20 DATA20,194,32,182,194,165,251,166,211,
   32,155,193,160,0,177,208       :rem  176
21 DATA133,210,177,212,72,144,13,200,177,
   208,72,177,212,136,145,212     :rem    6
22 DATA104,145,208,200,196,253,144,239,16
   5,2,208,3,169,32,44,165        :rem  130
23 DATA210,145,208,104,145,212,198,211,19
   8,254,16,201,32,20,194,88      :rem  224

24 DATA96,32,226,193,32,241,183,134,2,32,
   20,194,32,182,194,165          :rem   32
25 DATA251,166,211,32,155,193,177,208,133
   ,210,177,212,72,164,253,240    :rem   70
26 DATA15,136,177,208,72,177,212,200,145,
   212,104,145,208,136,208,241    :rem   65
27 DATA165,2,208,3,169,32,44,165,210,145,
   208,104,145,212,198,211        :rem  124
28 DATA198,254,16,203,48,68,166,252,138,2
   4,101,254,133,211,96,32        :rem  143
29 DATA226,193,160,0,177,122,201,44,208,8
   ,32,241,183,134,2,169          :rem   26

30 DATA0,44,169,255,133,174,32,20,194,32,
   182,194,165,251,166,211        :rem  128
31 DATA32,155,193,177,208,73,128,145,208,
   165,174,208,4,165,2,145        :rem  143
32 DATA212,136,16,239,198,211,198,254,16,
   226,32,20,194,88,96,32         :rem  100
33 DATA226,193,32,241,183,134,2,32,20,194
   ,32,182,194,165,251,166        :rem  129
```

LOAD



Supergraphic per C64

```
34 DATA211,32,155,193,165,2,145,212,136,1
   6,251,198,211,198,254,16       :rem  182
35 DATA236,48,215,32,226,193,32,241,183,1
   34,175,160,0,177,122,201       :rem  172
36 DATA44,208,8,32,241,183,134,2,169,0,44
   ,169,255,133,174,32            :rem  191
37 DATA20,194,32,182,194,165,251,166,211,
   32,155,193,165,175,145,208     :rem   37
38 DATA165,174,208,4,165,2,145,212,136,16
   ,241,198,211,198,254,16        :rem  142
39 DATA228,48,151,32,226,193,32,241,183,1
   34,175,160,0,177,122,201       :rem  176
40 DATA44,208,8,32,241,183,134,2,169,0,44
   ,169,255,133,174,32            :rem  186
41 DATA20,194,32,182,194,165,251,166,211,
   32,155,193,165,175,145,208     :rem   32
42 DATA165,174,208,4,165,2,145,212,136,16
   ,241,198,211,198,254,240       :rem  184
43 DATA228,48,34,165,251,166,211,32,155,1
   93,165,175,145,208,165,174     :rem   43
44 DATA208,4,165,2,145,212,136,48,4,160,0
   ,240,237,198,211,198           :rem  236
45 DATA254,240,194,16,222,32,20,194,88,96
                                  :rem   31
```





Supergraphic per C64
programma dimostrativo

```
0 PRINT"{CLR}":POKE53280,14:POKE53281,6
                                :rem  94
10 UP=12*4096+256:DWN=UP+67:LFT=UP+295:RH
   T=UP+369:RVS=UP+447:COL=UP+511
                                :rem  73
20 FILL=UP+547:BRDR=UP+611      :rem  63
32 GOTO35                       :rem   6
34 FORL=1TOND:NEXT:RETURN       :rem 208
35 PRINT"{WHT}{CLR}{ 11 GIU'}"TAB(13)"DIM
   OSTRAZIONE"                  :rem  68
38 PRINT"{GIU'}"TAB(13)"SUPERGRAPHIC!"
                                :rem 203
39 ND=25                        :rem 162
40 FORI=1TO4                    :rem 217
45 K=INT(RND(1)*8)              :rem 245
50 FORJ=0TO7                    :rem 221
60 SYS BRDR,J,J,40-2*J,25-2*J,160,J+K
                                :rem 243
65 GOSUB34                      :rem  82
70 NEXT                         :rem 166
75 K=INT(RND(1)*8)              :rem 248
80 FORJ=7TO0STEP-1              :rem 122
90 SYS BRDR,J,J,40-2*J,25-2*J,160,J+K
                                :rem 246
95 GOSUB34                      :rem  85
100 NEXT                        :rem 208
110 NEXT                        :rem 209
120 ND=1500:GOSUB34             :rem  73
130 PRINT"{CLR}":T=25:GOSUB1500:GOSUB34
                                :rem 203
135 ND=250                      :rem 255
140 FORI=1TO6                   :rem  12
150 SYS COL,4,14,31,3,15:GOSUB34:rem   6
160 SYS COL,4,14,31,3,6:GOSUB34 :rem 215
170 NEXT                        :rem 215
180 ND=1500:GOSUB34             :rem  79
190 SYS FILL,0,17,40,8,160,2    :rem 197
200 ND=750:GOSUB34              :rem  30
210 PRINT"{ 3 GIU'}{RED}{RVS}{ 7 DES}SCRO
    LL SENZA CANCELLAZIONE"     :rem 152
220 PRINTTAB(20)"{RVS}[< 13 T>]":rem  72
230 GOSUB34                     :rem 124
240 PRINT"{RVS}UTILIZZABILE IN TUTTE E 4
    LE DIREZIONI!!"             :rem  55
250 ND=1500:GOSUB34             :rem  77
260 PRINTTAB(17)"{RVS}{GIU'}SU" :rem 196
270 GOSUB34                     :rem 128
280 FORI=1TO22                  :rem  63
290 SYS UP,5,5,30,11,1          :rem 123
300 ND=75:GOSUB34               :rem 239
310 NEXT                        :rem 211
320 ND=750:GOSUB34              :rem  33
330 PRINTTAB(17)"{RVS}{SU}GIU'" :rem 166
340 GOSUB34                     :rem 126
350 FORI=1TO22                  :rem  61
360 SYS DWN,5,5,30,11,1         :rem 189
370 ND=75:GOSUB34               :rem 246
380 NEXT                        :rem 218
390 ND=750:GOSUB34              :rem  40
400 PRINTTAB(15)"{RVS}{SU}SINISTRA"
                                :rem   3
410 GOSUB34                     :rem 124
420 FORI=1TO40                  :rem  59
430 SYS LFT,0,5,40,11,1         :rem 180
440 ND=75:GOSUB34               :rem 244
450 NEXT                        :rem 216
460 ND=750:GOSUB34              :rem  38
470 PRINTTAB(15)"{RVS}{SU} DESTRA "
                                :rem  96
```

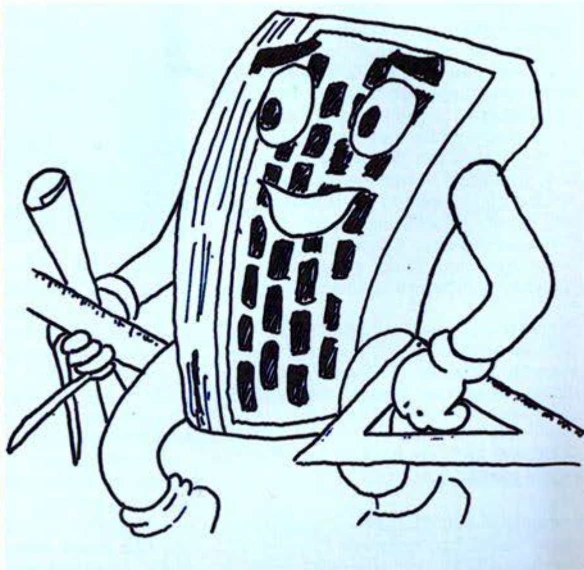
LOAD



Supergraphic per C64

```
480 GOSUB34                     :rem 131
490 FORI=1TO40                  :rem  66
500 SYS RHT,0,5,40,11,1         :rem 186
510 ND=75:GOSUB34               :rem 242
520 NEXT                        :rem 214
530 ND=750:GOSUB34              :rem  36
540 SYS FILL,0,17,40,8,160,5    :rem 199
550 ND=750:GOSUB34              :rem  38
560 PRINT"{ 5 SU}{RVS}{GRN}{DES}SCROLL CO
    N CANCELLAZIONE!"           :rem  22
570 PRINTTAB(12)"{RVS}[< 13 T>]":rem  81
580 GOSUB34                     :rem 132
590 PRINT"{ 2 DES}{RVS}{ 4 SPAZI}UTILE PE
    R CANCELLARE LO SCHERMO!"   :rem 188
600 GOSUB34                     :rem 125
610 PRINTTAB(17)"{RVS}COSI' ......."
                                :rem 161
620 GOSUB34                     :rem 127
630 FORI=1TO40                  :rem  62
640 SYS LFT,0,0,40,16,0         :rem 182
650 ND=0:GOSUB34                :rem 187
660 NEXT                        :rem 219
670 ND=750:GOSUB34              :rem  41
680 T=0:GOSUB1500               :rem 223
690 ND=750:GOSUB34              :rem  43
700 PRINTTAB(17)"{RVS}{ 8 GIU'}{GRN}O COS
    I' ........";               :rem 255
710 GOSUB34                     :rem 127
720 FORI=1TO40                  :rem  62
730 SYS DWN,0,0,40,17,0         :rem 186
740 ND=50:GOSUB34               :rem 240
750 NEXT                        :rem 219
760 ND=750:GOSUB34              :rem  41
770 PRINT"{CLR}":POKE53280,2:POKE53281,15
                                :rem 201
775 T=25                        :rem 155
780 PRINTTAB(9)"{GIU'}{BLK}UN TESTO PUO'
    ESSERE"                     :rem 134
790 PRINTTAB(14)"{GIU'}VISUALIZZATO{GIU'}
    "                           :rem 213
795 ND=750:GOSUB34              :rem  49
800 A$="SOLO IN UN UNICO":GOSUB2000:ND=75
    0:GOSUB34                   :rem 181
810 A$="{GIU'}COLORE SULLO SCHERMO.":GOSU
    B2000:ND=750:GOSUB34        :rem 100
820 PRINTTAB(3)"{ 2 GIU'}I COLORI POSSONO
     ESSERE CAMBIATI"           :rem  43
830 PRINTTAB(13)"{GIU'}IN SEGUITO...."
                                :rem 120
840 ND=1500:GOSUB34             :rem  82
850 SYS FILL,0,17,40,8,160,6    :rem 204
860 PRINT"{ 3 GIU'}{RVS}{BLU}{ 9 SPAZI}CO
    LORARE UNA FINESTRA."       :rem 152
870 PRINT"{ 8 DES}{RVS}[< 22 T>]"
                                :rem 108
875 ND=1500:GOSUB34             :rem  90
880 PRINT"{RVS}{ 4 SPAZI}CAMBIANDO IL COL
    ORE IN UNA"                 :rem 213
890 PRINT"{GIU'}{RVS} FINESTRA SI PUO' AT
    TRARRE L'ATTENZIONE"        :rem 129
900 ND=3000:GOSUB34             :rem  76
```

```
910 SYS COL,0,6,40,4,2              :rem  131
915 ND=1500:GOSUB34                 :rem   85
920 SYS FILL,0,20,40,5,160,6        :rem  193
925 ND=1500:GOSUB34                 :rem   86
930 PRINT"{ 3 SU}{RVS}{ 4 SPAZI}PUO' ANCH
     E ESSERE USATO{ 8 SPAZI}"      :rem  255
940 PRINT"{GIU'}{RVS}{ 5 SPAZI}PER FARLA
     LAMPEGGIARE.{ 9 SPAZI}"        :rem   42
945 ND=1500:GOSUB34:ND=250          :rem  248
950 FORI=1TO10                      :rem   64
955 SYS COL,0,6,40,4,15:GOSUB34     :rem  225
960 SYS COL,0,6,40,4,2:GOSUB34      :rem  169
970 NEXT                            :rem  223
975 ND=1500:GOSUB34                 :rem   91
976 SYS FILL,0,17,40,8,160,6        :rem  213
980 PRINT"{ 5 SU}{RVS}{BLU}{ 9 SPAZI}INVE
     RTIRE UNA FINESTRA{ 7 SPAZI}"
                                    :rem  112
990 PRINT"{RVS}{ 9 SPAZI}[< 22 T>]"
                                    :rem  135
995 ND=1500:GOSUB34                 :rem   93
1000 PRINT"{DES}{RVS}INVERTIRE I CARATTER
      I IN UNA FINESTRA"            :rem   56
1010 PRINT"{GIU'}{ 3 DES}{RVS}PUO' CREARE
      UN ALTO CONTRASTO"            :rem  106
1020 ND=3000:GOSUB34                :rem  118
1030 SYS RVS,3,11,33,5,5            :rem  255
1040 ND=1500:GOSUB34                :rem  123
1045 SYS FILL,0,20,40,5,160,6       :rem  240
1050 PRINT"{ 3 SU}{RVS}{ 11 SPAZI}PUO' AN
      CHE SERVIRE{ 10 SPAZI}"       :rem  246
1060 PRINT"{GIU'}{RVS}{ 9 SPAZI}PER FARLA
      LAMPEGGIARE.{ 8 SPAZI}"       :rem   84
1065 ND=1500:GOSUB34                :rem  130
1070 FORI=1TO14                     :rem  110
1075 SYS RVS,3,11,33,5,5:ND=200:GOSUB34
                                    :rem  196
1080 NEXT                           :rem    8
1085 ND=1500:GOSUB34                :rem  132
1090 SYS FILL,0,20,40,5,160,6       :rem  240
1100 PRINT"{ 3 SU}{RVS}{ 5 SPAZI}MA ....
      { 27 SPAZI}"                  :rem  158
1110 PRINT"{GIU'}{RVS}{ 5 SPAZI}PUO' ESSE
      RE DIVERTENTE...{ 9 SPAZI}":rem   29
1115 ND=1500:GOSUB34                :rem  126
1120 FORI=1TO60                     :rem  107
1125 SYS RVS,0,0,40,25:ND=120-.033*I*I:GO
      SUB34                         :rem   50
1130 NEXT                           :rem    4
1132 FORI=1TO20:SYS RVS,0,0,40,25:NEXT
                                    :rem  242
1135 ND=1500:GOSUB34                :rem  128
1145 PRINT"{CLR}":POKE53280,6:POKE53281,1
      5                             :rem  250
1150 V$="{HOME}{ 22 GIU'}"          :rem   75
1155 PRINTV$TAB(8)"{BLK}RIEMPIRE UNA FINE
      STRA"                         :rem  164
1160 ND=200:GOSUB34:SYS UP,0,0,40,24,0
                                    :rem   94
1165 PRINTV$TAB(8)"[< 21 T>]"       :rem  215
1170 GOSUB34:SYS UP,0,0,40,24,0 :rem  196
1175 GOSUB34:SYS UP,0,0,40,24,0 :rem  201
1180 PRINTV$TAB(6)"CON UN CARATTERE PARTI
      COLARE"                       :rem  199
1185 GOSUB34:SYS UP,0,0,40,24,0 :rem  202
1190 GOSUB34:SYS UP,0,0,40,24,0 :rem  198
1195 PRINTV$TAB(7)"CREA UN EFFETTO SIMPAT
      ICO!"                         :rem   15
1200 GOSUB34:SYS UP,0,0,40,24,0 :rem  190
1210 GOSUB34:SYS UP,0,0,40,24,0 :rem  191
1215 PRINTV$TAB(13)"SE IL CARATTERE"
                                    :rem  104
1220 GOSUB34:SYS UP,0,0,40,24,0 :rem  192
1225 GOSUB34:SYS UP,0,0,40,24,0 :rem  197
1230 PRINTV$TAB(10)"E' SCELTO CON CRITERI
      O!"                           :rem   50
1232 ND=3000:GOSUB34:ND=10          :rem  229
1235 FORI=1TO25:GOSUB34:SYS UP,0,0,40,24,
      0:NEXT                        :rem   33
1240 SYS FILL,5,4,33,15,160,2       :rem  243
1245 FORI=1TO7:SYS FILL,5,3+2*I,33,1,160,
      1:NEXT                        :rem  188
1250 SYS FILL,5,4,15,8,160,6        :rem  202
1255 FORI=1TO7:SYS FILL,4+2*I,5,1,6,170,6
      :NEXT                         :rem  148
1260 SYS FILL,2,3,1,20,160,0        :rem  182
1270 PRINTV$TAB(13)"DIVERTENTE !!"
                                    :rem  221
1499 GOTO1499                       :rem  231
1500 PRINT"{HOME}":POKE53280,12:POKE53281
      ,15                           :rem  162
1510 PRINTTAB(10)"{BLK}[< 20 @>]"
                                    :rem  129
1520 PRINTTAB(10)"{RVS} COMANDI DI SCROLL
      . {OFF}"                      :rem   75
1530 ND=60*T:GOSUB34                :rem  157
1540 A$="QUESTI COMANDI":PRINT"[<7>]
      {GIU'}":GOSUB2000:ND=20*T:GOSUB34
                                    :rem  194
1550 A$="OFFRONO PARECCHIE POSSIBILITA'":
      PRINT"{GRN}{GIU'}":GOSUB2000:GOSUB34
                                    :rem  143
1560 A$=" PER CREARE ":PRINT"[<1>]{GIU'}"
      :GOSUB2000:ND=20*T:GOSUB34 :rem  110
1570 A$="S U P E R{ 3 SPAZI}E F F E T T I
       ! ! ":PRINT"{BLU}{GIU'}":GOSUB2000:
      ND=20*T:GOSUB34               :rem   76
1580 RETURN                         :rem  174
2000 PRINTTAB((40-LEN(A$))/2);      :rem  138
2005 ND=T                           :rem  234
2010 FORI=1TOLEN(A$)                :rem  153
2020 PRINTMID$(A$,I,1)"{RVS} {OFF}{SIN}";
                                    :rem  151
2030 GOSUB34                        :rem  172
2040 NEXT                           :rem    5
2050 PRINT" "                       :rem  152
2060 RETURN                         :rem  168
```





Caverne spaziali

di **V. Valenti**
trad. ed adatt.
di **M. Cristuib Grizzi**

Ecco un gioco stile "arcade" in cui dovete combattere contro terribili alieni: per C64 e VIC 20 con espansione di memoria da almeno 8 Kbyte. È richiesto l'uso di un joystick.

L'obiettivo di "Caverne spaziali" consiste nel cercare di evitare alcuni alieni che cercano di atterrare sulla vostra nave spaziale a forma di diamante. Potete muovere la nave nelle quattro direzioni, su, giù, a destra e a sinistra, ma non esiste via di fuga dalle pericolose caverne: potete solo avanzare ulteriormente nelle loro profondità, sparando agli alieni prima di venirne sopraffatti.

Gli alieni, come se non bastasse, non sono stupidi: le creature del pianeta Kung sono molto intelligenti e vi raggiungeranno probabilmente più in fretta di quanto voi sarete in grado di fuggire. Se vi organizzate in modo tale da sconfiggerli e distruggerli in un certo numero di caverne successive, otterrete un round in bonus, nel quale sarete in contatto solamente con alieni... sedentari che appaiono a caso sullo schermo.

Un orologio limita l'ammontare di tempo disponibile per catturare questi alieni e, quando il tempo finisce, avanzate ancora nelle caverne ad un livello più elevato di difficoltà nel gioco.

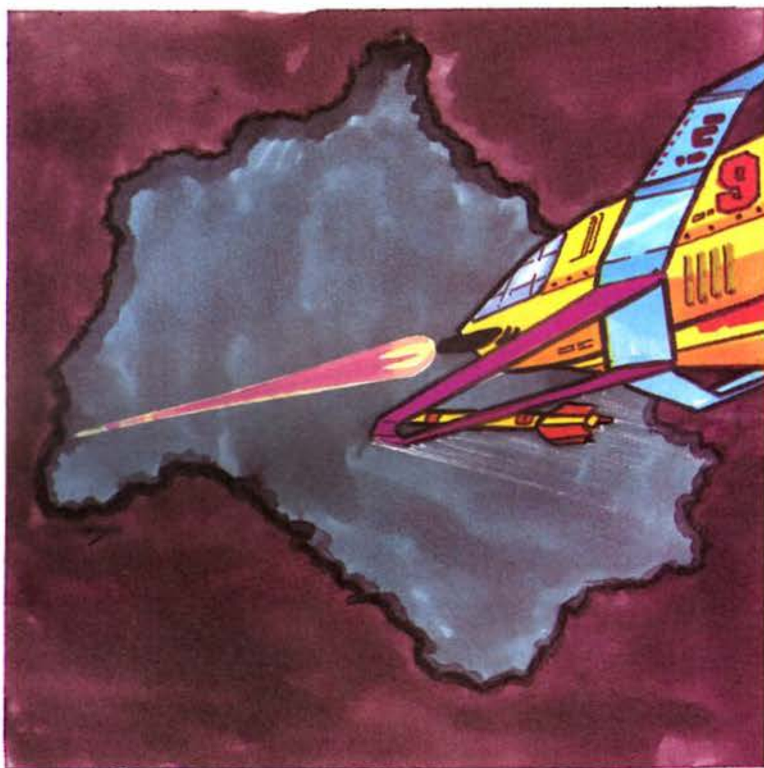
Dopo ogni gioco i tre migliori punteggi lampeggeranno sullo schermo, invogliandovi a superare le vostre prestazioni.

Prima di caricare il programma sul VIC 20 assicuratevi di aver inserito l'espansione di memoria da almeno 8 Kbyte e digitate le seguenti istruzioni in modo diretto:

```
POKE44,28:POKE43,1:
POKE28*256,0:NEW
```

premete quindi RETURN e caricate come al solito il programma.

Nessuna POKE particolare è invece richiesta dalla versione per C64, il quale utilizza per il joystick la Porta 2.



Entrambe le versioni sono simili per quanto riguarda la struttura del gioco: il numero di caverne è sempre il doppio del livello di gioco, ed i punti guadagnati nella cattura degli alieni "stazionari" durante il round di bonus sono proporzionali al tempo segnato dall'orologio. Inoltre, ogni caverna possiede una piccola grotta al centro dello schermo. Nella versione per VIC 20 la vostra nave appare sempre all'interno di questa grotta: non esiste uscita, ma potete aprire delle aperture nelle pareti sparando con il vostro cannone laser. Le pareti, tuttavia, non saranno degli ostacoli per gli alieni.

Nella versione per C64 la vostra nave appare in una posizione casuale sullo schermo – all'inizio di ogni caverna – a volte all'interno della grotta centrale, ed a volte al suo esterno. Se la nave appare all'interno della grotta, dovrete fuggirne velocemente, dal momento che il vostro laser non funzionerà fino a che ne sarete usciti.

Per sparare spostate il joystick nella direzione desiderata e premete il pulsante di fuoco.

Caverne spaziali
versione per C64

```
10 GOSUB1300                              :rem 165
20 V=54272:POKEV+24,15:POKEV+5,0:POKEV+6,
   240:POKEV,100                          :rem 180
30 POKE53281,0:POKE53280,0                :rem 184
40 PRINT"{CLR}{ 8 GIU'}{ 13 DES}[<1>]CAVE
   RNE SPAZIALI"                          :rem 171
50 PRINT"{ 3 GIU'}{ 14 DES}[<7>]HIGH SCOR
   ES"                                    :rem 136
60 PRINTTAB(17)"{GIU'}{RED}"H1            :rem 109
70 PRINTTAB(17)"{GIU'}{PUR}"H2            :rem 239
80 PRINTTAB(17)"{GIU'}{GRN}"H3            :rem 115
90 PRINT"{YEL}{ 2 GIU'}{ 5 DES}PREMI FUOC
   O PER GIOCARE"                         :rem 101
100 IFPEEK(56320)AND16THEN100             :rem 144
110 L=1:S=0:R=0:SH=3:Q=0:W=33             :rem 176
120 IFR=L*2THENL=L+1:R=0:GOSUB1040
                                          :rem 49
130 PRINT"{CLR}{ 11 GIU'}"TAB(16)"{RED}LI
    V. :{WHT}"L                           :rem 70
140 PRINT"{ 7 GIU'}"TAB(16)"{GRN}CAV.:
    {WHT}";R+1                            :rem 254
150 FORI=1TO1000:NEXT                     :rem 17
160 POKE53281,11:PRINT"{CLR}":POKE53281,0
                                          :rem 189
170 IFS>10000ANDQ=0THENPRINT"{HOME}
    { 15 DES}{WHT}NAVE EXTRA":SH=SH+1:Q=1
                                          :rem 181
180 Y1=L+1:X1=L:Y2=L+1:X2=39-L:X3=39-L:Y3
    =22-L:Y4=22-L:X4=L                    :rem 219
190 A1=1:A2=1:A3=1:A4=1                   :rem 206
200 FORI=0TO39:POKEI+1064,37:POKEI+1904,3
    7:NEXT                                :rem 54
210 FORI=5TO34:POKEI+1224,37:POKEI+1744,3
    7:NEXT                                :rem 55
220 FORI=12TO27:POKEI+1384,37:POKEI+1584,
    37:NEXT                               :rem 113
230 POKE1244,32:POKE1404,32:POKE1764,32:P
    OKE1604,32                            :rem 116
240 FORI=1TO22:POKEI*40+1024,37:POKEI*40+
    1063,37:NEXT                          :rem 71
250 FORI=5TO18:POKEI*40+1029,37:POKEI*40+
    1058,37:NEXT                          :rem 90
260 FORI=9TO14:POKEI*40+1036,37:POKEI*40+
    1051,37:NEXT                          :rem 82
270 POKE1509,32:POKE1516,32:POKE1538,32:P
    OKE1531,32                            :rem 126
280 X=INT(RND(1)*10)+15:Y=INT(RND(0)*6)+8
                                          :rem 33
290 PRINT"{HOME}{ 40 SPAZI}"              :rem 127
300 POKE1024+X+40*Y,W:POKE55296+X+40*Y,1
                                          :rem 7
310 IFX=X1ANDY=Y1THEN910                  :rem 253
320 IFX=X2ANDY=Y2THEN910                  :rem 0
330 IFX=X3ANDY=Y3THEN910                  :rem 3
340 IFX=X4ANDY=Y4THEN910                  :rem 6
350 W=W+1:IFW=35THENW=33                  :rem 205
360 POKE1024+X1+40*Y1,W+2:POKE55296+X1+40
    *Y1,5                                 :rem 50
370 POKE1024+X2+40*Y2,W+2:POKE55296+X2+40
    *Y2,5                                 :rem 55
380 POKE1024+X3+40*Y3,W+2:POKE55296+X3+40
    *Y3,5                                 :rem 60
390 POKE1024+X4+40*Y4,W+2:POKE55296+X4+40
    *Y4,5                                 :rem 65
400 PRINT"{HOME}";:POKE211,5:PRINT"PUNTI:
    ";S;:POKE211,30:PRINT"NAVI :";SH
                                          :rem 132
410 IF(A1+A2+A3+A4)=0THEN850              :rem 253
420 J=PEEK(56320):D=JAND15:B=JAND16
                                          :rem 14
430 IFD=15THEN680                         :rem 218
440 IFX>12ANDX<27ANDY>9ANDY<14THEN460
                                          :rem 41
450 IFB=0THEN540                          :rem 159
460 POKE1024+X+40*Y,32                    :rem 182
470 XA=0:YA=0                             :rem 226
480 IFD=14ANDY>L+1THENYA=-1               :rem 135
490 IFD=13ANDY<22-LTHENYA=1               :rem 141
500 IFD=11ANDX>LTHENXA=-1                 :rem 31
510 IFD=7ANDX<39-LTHENXA=1                :rem 95
520 IFPEEK(1024+(X+XA)+(Y+YA)*40)=32THENX
    =X+XA:Y=Y+YA                          :rem 147
530 GOTO680                               :rem 111
540 POKEV+4,129:FORI=100TO10STEP-10:POKEV
    +1,I:NEXT:POKEV+4,128                 :rem 57
550 IFD=14THENJ=X:FORI=Y-1TO1STEP-1:GOSUB
    600:IFBTHENNEXT                       :rem 178
560 IFD=13THENJ=X:FORI=Y+1TO22:GOSUB600:I
    FBTHENNEXT                            :rem 73
570 IFD=11THENI=Y:FORJ=X-1TO1STEP-1:GOSUB
    600:IFBTHENNEXT                       :rem 177
580 IFD=7THENI=Y:FORJ=X+1TO39:GOSUB600:IF
    BTHENNEXT                             :rem 38
590 GOTO680                               :rem 117
600 B=-1                                  :rem 115
610 IFJ=X1ANDI=Y1THENA1=0:X1=0:Y1=0:S=S+5
    0:B=0                                 :rem 30
620 IFJ=X2ANDI=Y2THENA2=0:X2=1:Y2=0:S=S+1
    00:B=0                                :rem 81
630 IFJ=X3ANDI=Y3THENA3=0:X3=2:Y3=0:S=S+1
    50:B=0                                :rem 93
640 IFJ=X4ANDI=Y4THENA4=0:X4=3:Y4=0:S=S+2
    00:B=0                                :rem 96
650 IFPEEK(1024+I*40+J)<>32THENB=0
                                          :rem 155
660 POKE55296+I*40+J,3:POKE1024+I*40+J,39
    :FORK=1TO10:NEXT:POKE1024+I*40+J,32
                                          :rem 122
670 RETURN                                :rem 125
680 IFA1=0THEN720                         :rem 212
690 POKE1024+X1+40*Y1,32                  :rem 29
700 X1=X1+(X1>X)-(X1<X)                   :rem 28
710 Y1=Y1+(Y1>Y)-(Y1<Y)                   :rem 35
720 IFA2=0THEN760                         :rem 212
730 POKE1024+X2+40*Y2,32                  :rem 26
740 X2=X2+(X2>X)-(X2<X)                   :rem 36
750 Y2=Y2+(Y2>Y)-(Y2<Y)                   :rem 43
760 IFA3=0THEN800                         :rem 212
770 POKE1024+X3+40*Y3,32                  :rem 32
780 X3=X3+(X3>X)-(X3<X)                   :rem 44
790 Y3=Y3+(Y3>Y)-(Y3<Y)                   :rem 51
800 IFA4=0THEN840                         :rem 212
810 POKE1024+X4+40*Y4,32                  :rem 29
820 X4=X4+(X4>X)-(X4<X)                   :rem 43
830 Y4=Y4+(Y4>Y)-(Y4<Y)                   :rem 50
840 GOTO300                               :rem 104
850 N=N+1:PRINT"{HOME}{ 5 DES}
    { 10 SPAZI}{WHT}CAV.SUCCESSIVA
    { 11 SPAZI}"                          :rem 135
860 FORG=0TOX:POKE55296+G+40*Y,6:POKE1024
    +G+40*Y,38:NEXT                       :rem 82
870 FORG=X-1TO0STEP-1:POKE55296+G+Y*40,1:
    POKE1024+G+Y*40,33                    :rem 200
880 POKE1025+G+Y*40,32:NEXT               :rem 37
890 PRINT"{CLR}{ 11 GIU'}{ 14 DES}{CYN}BO
    NUS:{RED}";L*500+R*50:S=S+L*500+R*50:
    R=R+1                                 :rem 250
900 FORI=1TO1500:NEXT:GOTO120             :rem 31
910 POKEV+1,50:POKEV+4,129:X1=X-1:X2=X+1:
    Y1=Y-1:Y2=Y+1                         :rem 204
920 FORI=1TO20                            :rem 62
```

ECCO DOVE PUOI CONSULTARE TUTTA LA BIBLIOTECA DEI JACKSO



ELETTRONICA E INFORMATICA DI BASE
HOME & PERSONAL COMPUTER

ENCICLOPEDIE E DIZIONARI
GUIDE E MANUALI TECNICI



IANI





GRUPPO EDITORIALE JACKSON
Milano-San Francisco-Londra-Madrid

PIEMONTE

ALESSANDRIA
Libreria Bertolotti
Corso Roma, 122

BIELLA (VC)
Libreria Giovannacci
Via Italia, 14

CHIVASSO (TO)
Cooperativa di Cultura "L. Milani"
Via Teodoro II, 12

CUNEO
Libreria L'Ippogrifo
Piazza Europa, 3

DOMODOSSOLA (NO)
Libreria Rizzardi
Corso Moneta 1

FOSSANO (CN)
Libreria Bosatti Franco
Via Roma, 149

NOVARA
Casa del Libro
Via F.lli Rosselli, 45

PINEROLO (TO)
Libreria Elia Romano
Piazza Vittorio Veneto, 4

Libreria Gianoglio di Combetto Gian Luigi
Via Duomo, 11

TORINO
Libreria Dante Alighieri di Fogola
Piazza C. Felice, 15

Libreria Ape d'Oro
Corso Francia, 35

Cartolibreria dell'ITIS "Avogadro"
Corso S. Maurizio, 31

CAMPUS Libri
Via U. Rattazzi, 4

CELID - Sezione Politecnico
Corso Duca degli Abruzzi, 24

Cooperativa Studi - C.S.
Via Ormea, 67 bis

Libreria Scientifica Universitaria Cortina
Corso Marconi, 34 A

Libreria Druetto
Via Roma, 227

Galleria del Libro
Via Maria Vittoria, 1

Libreria Hellas
Via Bertola, 6

Libreria Lattes
P.zza Garibaldi, 3

Libreria Orsa Maggiore
Corso G...lio Cesare, 58

Libreria Petrini
Via P. Micca, 22

VERBANIA INTRA (NO)
Libreria Alberti
C.so Garibaldi 74

LIGURIA

GENOVA
Libreria Bozzi
Via Cairoli

Libreria Di Stefano
Piazza Fontane Marose

Libreria Di Stefano Tecnica
Piazza Ceccardi, 40

Fiera del libro
Via Porta d'Archi 10r

Banco Libri Rimondi
Via Dante - Portici Posta

GENOVA SAMPIERDARENA
Libreria Gaggiolo
Piazza V. Veneto, 29r

SAVONA
Libreria G.B. Moneta
Via Boselli, 8r

Libreria Moneta Vittorio
Corso Italia, 71r

LOMBARDIA

BERGAMO
Libreria Antica e Moderna di E. Lorenzelli
Viale Papa Giovanni XXIII, 74

Bergamolibri
Via Palazzolo, 21

Libreria Rasmussen
Via Sauri, 4

BRESCIA
Libreria Bonzoni di Vorrasi
Via Trieste, 32

Libreria Dolcassi
Via Trieste, 25/B

Libreria Pavoniana - Edizioni Ancora
Via Tosio, 1

Libreria Resola
Corso Garibaldi, 39

BUSTO ARSIZIO (VA)
Libreria Rinascita Centro Culturale
Via Foscolo, 2

CINISELLO BALSAMO (MI)
La Libreria
Via Lincoln, 46

COMO
Mondadori
Via Vittorio Emanuele, 36

LODI (MI)
Libreria Castello
Corso Vittorio Emanuele, 31

MANTOVA
Libreria ERRE EMME
Via Accademia, 31

MERATE (CO)
Libreria LA "TORRE"
Via Don Cazzaniga, 6

MILANO
Libreria Del Convegno
Via Lomellina, 35

Libreria DEL CORSO di Giagiannello Franco
Corso Buenos Aires, 49

Libreria EUROPA
Via S. Tecla, 5

I.P.S.O.A.
Largo Augusto, 8

Libreria "LA TRAMITE"
Piazza Medaglie d'Oro, 3

"LIBROPOLI"
Via Arconati, 16

Libreria "MARCO" Sedis
Galleria Passarella, 2

Libreria Paravia
Corso Matteotti, 3

Libreria Internazionale "PARTIPILO"
Viale Tunisia, 4

Libri Remainders Center
Corso Buenos Aires, 42/3

PIACENZA
Libreria Del Teatro
Via Verdi, 5

VARESE
Libreria Pontiggia
Corso A. Moro, 3

TRE VENEZIE

BASSANO GRAPPA (VI)
Libreria Scrimin
Piazza Garibaldi, 25

BELLUNO
Libreria Mossenz
Via Matteotti, 58

BOLZANO
Libreria Cappelli
P.zza Vittoria, 41

Tuttolibri
Via Roma, 7/B

FELTRE (BL)
Libreria Moderna W. Pilotto

LIDO DI JESOLO (VE)
Libreria Gianese Giuseppe
Via Bafile, 87

MESTRE (VE)
La libreria di Via Bissuola di Catano Marinella
Via Bissuola, 12/14

Libreria Pacinotti
Via Canove, 94

PADOVA
Libreria Internazionale CORTINA
Via Marzolo, 2

Libreria Nuova Moderna
Via Paolotti, 5

PORDENONE
Cartolibreria San Giorgio
Via XXX Aprile

PORTOGRUARO (VE)
Cartolibreria Biasutti di Fanini Antonio & C. s.n.c.
Corso Martiri della Libertà, 160

TRENTO
Libreria Artigianelli
Via S. Croce

UNIVERSITARIA
Via Travai, 28

TREVISO
CANOVA
Società cartolibraria editrice
Via Calmaggiore, 31

TRIESTE
Libreria Internazionale Borsatti
Via Dante, 14

Libreria Italo Svevo
Corso Italia

Libreria Tergeste
Piazza Borsa, 15

UDINE
Libreria Carducci di Feruglio dott. Fausto & C.
Piazza XX Settembre, 16

Coop. Libraria "Borgo Aquileia"
Via Aquileia, 53

VENEZIA
Libreria SOLARIS
Via Canareggio, 2332

VERONA
Libreria Cangrande
Via IV Novembre, 22

Libreria Gheffi Barbato
Via Mazzini, 21

Libreria GROSSO
Via Carducci, 144

Marchiotto
Via del Pontiere, 10

EMILIA ROMAGNA

BOLOGNA
Libreria Rizzoli
Via Rizzoli, 8

Libreria Zanichelli
Piazza Galvani

CARPI (MO)
Libreria Rinascita
Piazza Martiri, 50

CESENA (FO)
Libreria Bettini
Via Vescovado

FERRARA
Centro C. Informazione
Via S. Stefano, 54

Spazio libri
Via del Turco

MODENA
Libreria Galileo
Via Emilia

ECCO DOVE PUOI CONSULTARE TUTTA LA BIBLIOTECA DEI JACKSONIANI



TOSCANA

AREZZO
Libreria Mori
Via Roma 24

Pinocchio
Via Vitt. Veneto 12

FIRENZE
Libreria Feltrinelli
Via Cavour, 12

Libreria le Monnier
Via S. Gallo, 49r

Libreria Marzocco
Via de' Martelli, 6

Libreria Pirola
Via Cavour, 46r

LIVORNO
Amedeo Nuova Libreria
Corso Amedeo, 23

Libreria Belforte
Via Grande, 91

LUCCA
Sestante
Via Montanara, 9

PISA
C.L.U.
Via S. Maria, 7

Libreria Feltrinelli
Corso Italia, 117

Ghibellina
Borgo Stretto, 37

Testi Universitari
Via S. Maria, 14

Libreria Vallerini
Lungarno Pacinotti, 10

POGGIBONSI (SI)
Mickey Mouse
Via Trento, 50

PONTEDERA (PI)
Libreria Carrara
Corso Matteotti, 103

PRATO (FI)
Libreria Gori
Via Ricasoli, 26

La luna
Via dell'Altopascio, 21/23/25

SESTO FIORENTINO (FI)
Libreria Rinascita
Via Gramsci, 334

SIENA
Libreria Ticci
Via delle Terme, 5/7

UMBRIA

PERUGIA
Libreria BETTI
Corso Vannucci

TERNI
Libreria Alterocca
Corso Tacito

MARCHE

ANCONA
Libreria Canonici
Corso Garibaldi, 132

FABRIANO (AN)
Libreria Babele
Corso Repubblica, 61

FERMO (AP)
Galleria del libro
Piazza del Popolo, 24

MACERATA
Bottega del libro
Corso Repubblica, 7

PESARO
FIM LIBRO
Via Abbati, 39

RECANATI (MC)
Libreria dell'incontro
Via Calcagni, 3

S. BENEDETTO DEL TRONTO (AP)
Libreria Nuovi Orizzonti
Via Montebello, 61

LAZIO

ROMA
Libreria Arethusa
Viale della Primavera, 93

Libreria "Asterisco"
Via dei Colli Portuensi, 379

Libreria "Aurora"
Piazza Mazzini, 22

Libreria "Calliope"
Via S. Croce in Gerusalemme, 89/b

Libreria Coletti a S. Pietro
Largo del Colonnato, 5

Libreria Internazionale Croce
Via Solferino, 7/ab

Libreria De Miranda
Via G. Cesare, 51e

Libreria Gremese
Via Cola di Rienzo, 136

Libreria Moraldi
Via Leone IV, 7

Libreria Micozzi
Via G. Ferrari, 39/41

Mondadori per voi
Via Nazionale, 246

Libreria "Palma"
Viale Europa, 1

Libreria Panella
Piazza dei Re di Roma, 66

Libreria Internazionale Rizzoli
Largo Chigi

Libreria S. Silvestro
Piazza S. Silvestro, 27/28

Self Service Dei Libro
Via Terme di Diocleziano, 35/36

Libreria Tutti Libri
Via Appia Nuova, 447

Libreria Universitaria
Ingegneria 2000
Via della Polveriera, 15

ABRUZZO E MOLISE

CAMPOBASSO
Libreria Giuridica di E.M.
Via Monsignor Bologna, 67

LANCIANO (CH)
Cartolibreria Cipolla
Via O. Bocache, 6

TERAMO
Libreria LA SCOLASTICA
Corso S. Giorgio, 39

CAMPANIA

AVELLINO
Libreria C.E. Menna
Via Vasto, 15/19

Libreria Petrella s.r.l.
Corso V. Emanuele, 133

BENEVENTO
Cooperativa Nuovo Politecnico
Via Capilongo, 14/16

Libreria De Luca Concetta
Piazza S. Sofia, 23

NAPOLI
C.U.E.N.
Piazzale Tecchio (Facoltà Ingegneria)

Deperro Libri s.r.l.
Via dei Mille, 17/19

Libreria "A. Guida s.r.l."
Via Merliani, 118

Libreria "A. Guida s.r.l."
Via Port'Alba, 20/3

Cartolibreria L'Ateneo di Pironti
Viale Augusto, 168/170

Libreria Commissionaria Liguori s.n.c.
Via Mezzocannone, 21/23

Libreria L'Internazionale di Vincenzo Morra
Via Scarlatti, 149

Libreria Loffredo Luigi
Via Kerbaker, 19/21

Libreria Pironti Tullio s.r.l.
Piazza Dante, 30/31

PISANTI Renato s.r.l.
Libreria Scientifica Editrice
Corso Umberto I, 38/40

Punto Quattro s.n.c.
Via G. Cesare, 22/23

Supermercato del Fumetto di Siena Riccardo
Via Montesanto, 14

PENTA-FISCIANO (SA)
Universitaria Liguori
Via S. Rocco

SALERNO
Libreria Carrano Umberto
Via dei Mercanti, 53

Libreria Paolillo Rodolfo
Via dei Principati, 44/46

PUGLIA

BARI
Libreria Cooperativa
Via Crisanzio, 12

G.B.C.
Via Capruzzi, 192

Libreria Laterza
Via Sparano, 134

FOGGIA
Libreria DANTE
Via Oberdan, 1

LECCE
Mostra del Libro di Monaco C.
Viale Marconi, 9

MATERA
Libreria Cifarelli
Piazza V. Veneto, 43/45

MOLFETTA (BA)
Il Ghigno
Via Salepico, 37

CALABRIA

LOCRI (RC)
Libreria Francesco Pancallo
Corso V. Emanuele, 93

Libreria Francesco Pedullà
Corso V. Emanuele, 80

REGGIO CALABRIA
Libreria Scientifica
Corso Garibaldi, 154
(Galleria Caminiti)

SICILIA

CATANIA
Libreria Bonaccorso Carmelo
Via Etnea, 20/22

Libreria C.C. Cavallotto
Corso Sicilia, 91

Libreria Crisafulli
Via Etnea, 280

Libreria Editrice Giannotta di Sebastiano Paco Giannotta
Viale R. Margherita, 2/ef

Libreria Minerva di Carmelo Tringale
Corso Italia, 19/21

Libreria Patti di Franco Marchese Patti
Via V. Emanuele, 294/296

Libreria Scientifica di A. Marino
Piazza Trento, 3/D

ENNA
Cartolibreria MINERVA di Mario Castellana
Via Roma, 383

GIARRE (CT)
Libreria LA SENORITA di Pina Emmi
Corso Italia, 132

LENTINI (SR)
Libreria Amore Inserra
Via Garibaldi, 15

MESSINA
Libreria Editrice HOBELIX
Soc. Coop. R.L.
Via dei Verdi, 21

Cartolibreria Giuseppe Prinzi di G. & L. Prinzi
Via T. Cannizzaro, 161-165

Cartolibreria Giuseppe Prinzi di Prinzi Antonino
Viale S. Martino, 254

PALERMO
Libreria CAVOUR
Via Cavour, 116

C.E.L.U.P.
Via Carducci, 3

Libreria DEL CORSO di Profita Mario
Via Turati, 15

Libreria Flaccovio Dario Editore
Via Ausonia, 70/74

Libreria Flaccovio Dario Editore
Via Campolo, 76

Libreria S.F. Flaccovio
Via E. Basile, 136

Libreria S.F. Flaccovio
Piazza Don Bosco, 3

Libreria S.F. Flaccovio
Quattro Canti di Città (Dante)

Libreria S.F. Flaccovio
Piazza V.E. Orlando, 14

Libreria S.F. Flaccovio
Via R. Settimo, 37

Libreria PAPER C
Via Laurana, 89-91

Libreria PAPER C
Viale Lazio, 83

Libreria PAPER C
Via Libertà, 15

Libreria PAPER C
Piazza Politeama, 6

Libreria PAPER C
Via Sciuti, 140/ABC

Libreria Edizione Pegaso
Via Notarbartolo, 9/F

Testo Universitario
Viale Lazio, 60/B

SIRACUSA
Libreria del Corso di R. Toccoli
Corso Umberto, 116

VITTORIA (RG)
Libreria SAPERE del Dr. Salvatore Marangio
Via Cavour, 80

SARDEGNA

CARBONIA (CA)
Libreria D.E.S.I.
Via Dalmazia

SVIZZERA ITALIANA

LUGANO **Canton Ticino**
Melisa
Via Vegezzi, 4

Le librerie indicate hanno partecipato alla campagna promozionale per la "Biblioteca dei Jacksoniani". Qualora altre librerie, non menzionate, desiderassero aderire a iniziative analoghe sono pregate di prender contatto con:

DISTRIBUZIONE-PETRINI EDITORE S.P.A.
10139 Torino - C.so Trapani, 48
Tel. 011-3358641

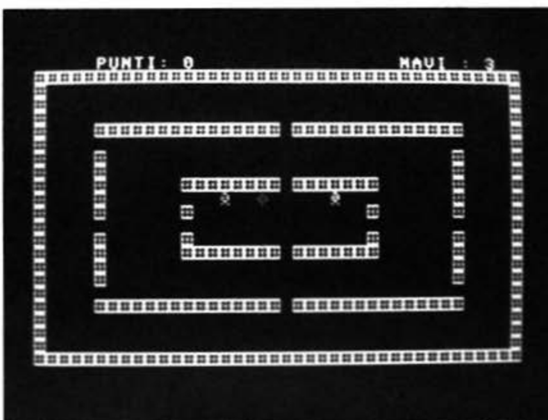


GRUPPO EDITORIALE JACKSON
Milano-San Francisco-Londra-Madrid

Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951-5 linee

Noi l'informatica la conosciamo davvero

```
930 POKE1024+X1+40*Y1,38:POKE55296+X1+40*
    Y1,2                              :rem 233
940 POKE1024+X2+40*Y1,38:POKE55296+X2+40*
    Y1,2                              :rem 236
950 POKE1024+X1+40*Y2,38:POKE55296+X1+40*
    Y2,2                              :rem 237
960 POKE1024+X2+40*Y2,38:POKE55296+X2+40*
    Y2,2                              :rem 240
970 IFX1>0THENX1=X1-1                 :rem 2
980 IFX2<39THENX2=X2+1                :rem 62
990 IFY1>0THENY1=Y1-1                 :rem 7
1000 IFY2<22THENY2=Y2+1               :rem 89
1010 NEXT:POKEV+4,128                 :rem 230
1020 SH=SH-1:IFSH=0THEN1250           :rem 92
1030 GOTO120                          :rem 144
1040 T=200+L*60:X=20:Y=11             :rem 82
1050 POKE53281,11:PRINT"{CLR}":POKE53281,
     0                                :rem 236
1060 PRINT"{HOME}{ 14 DES}{GRN}ROUND BONU
     S"                               :rem 110
1070 FORI=1064TO1103:POKEI,37:POKEI+21*40
     ,37:NEXT                         :rem 141
1080 FORI=1064TO1064+40*21STEP40:POKEI,37
     :POKEI+39,37:NEXT                :rem 203
1090 POKEV+1,5:POKEV+4,33:FORI=10TO100STE
     P2:POKEV+1,I:NEXT:POKEV+4,32
                                      :rem 30
1100 X1=INT(RND(0)*40)                :rem 183
1110 Y1=INT(RND(0)*20)+1:IFPEEK(1024+X1+Y
     1*40)<>32THEN1100                :rem 224
1120 POKE55296+X1+Y1*40,6:POKE1024+X1+Y1*
     40,36:GOTO1140                   :rem 76
1130 IFPEEK(1024+X+Y*40)=36THENS=S+T:GOTO
     1100                             :rem 146
1140 POKE55296+X+Y*40,1:POKE1024+X+Y*40,W
                                      :rem 58
1150 W=W+1:IFW=35THENW=33             :rem 252
1160 PRINT"{HOME}{PUR}PUNTI:{WHT}";S;:POK
     E211,30:PRINT"{YEL}TEMPO:{WHT}";T;"
     {SIN} "                          :rem 132
1170 J=PEEK(56320):D=JAND15:B=JAND16
                                      :rem 65
1180 POKE1024+X+40*Y,32               :rem 230
1190 IFD=14ANDY>2THENY=Y-1            :rem 88
1200 IFD=13ANDY<21THENY=Y+1           :rem 124
1210 IFD=11ANDX>1THENX=X-1            :rem 74
1220 IFD=7ANDX<38THENX=X+1            :rem 86
1230 T=T-1:IFT=0THENRETURN            :rem 162
1240 GOTO1130                         :rem 197
1250 IFS>H1THENH3=H2:H2=H1:H1=S:GOTO1280
                                      :rem 173
1260 IFS>H2THENH3=H2:H2=S:GOTO1280
                                      :rem 70
1270 IFS>H3THENH3=S                   :rem 159
1280 PRINT"{CLR}{ 14 GIU'}{ 15 DES}{RED}G
     AME OVER"                        :rem 66
1290 FORI=1TO1500:NEXT:GOTO30         :rem 34
1300 PRINT"{CLR}{ 11 GIU'}{ 9 DES}STO MOD
     IFICANDO I CARATTERI..."         :rem 109
1310 POKE56334,PEEK(56334)AND254:POKE1,PE
     EK(1)AND251                      :rem 230
1320 FORI=0TO511:POKE12288+I,PEEK(53248+I
     ):NEXT                           :rem 20
1330 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
     4)OR1                            :rem 182
1340 I=12552                          :rem 77
1350 READA:IFA=-1THEN1370             :rem 197
1360 POKEI,A:I=I+1:GOTO1350           :rem 80
1370 POKE53272,(PEEK(53272)AND240)OR12
                                      :rem 96
1380 RETURN                           :rem 172
1390 DATA 0,8,20,34,73,34,20,8        :rem 183
1400 DATA 0,0,8,20,42,20,8,0          :rem 61
1410 DATA 24,60,106,126,102,60,24,60
                                      :rem 213
1420 DATA 24,60,86,126,102,60,24,102
                                      :rem 218
1430 DATA 255,219,165,219,219,165,219,255
                                      :rem 246
1440 DATA 136,33,0,148,1,72,2,16:rem 23
1450 DATA 0,0,0,24,24,0,0,0           :rem 4
1460 DATA -1                          :rem 67
```



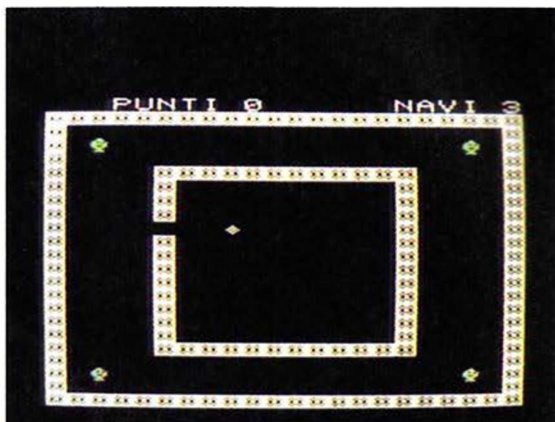
Caverne spaziali
versione per VIC 20

```
10 GOSUB1270                          :rem 171
20 POKE36878,15                       :rem 51
30 POKE36879,8                        :rem 7
40 PRINT"{CLR}{ 6 GIU'}{ 4 DES}{WHT}CAVER
   NE SPAZIALI"                       :rem 8
50 PRINT"{ 3 GIU'}{ 5 DES}{BLU}HIGH SCORE
   S"                                 :rem 8
60 PRINTTAB(8)"{GIU'}{RED}"H1         :rem 61
70 PRINTTAB(8)"{GIU'}{PUR}"H2         :rem 191
80 PRINTTAB(8)"{GIU'}{GRN}"H3         :rem 67
90 PRINT"{YEL}{ 2 GIU'}{ 2 DES}FUOCO PER
   GIOCARE"                           :rem 145
100 IFPEEK(37137)AND32THEN100         :rem 147
110 L=1:S=0:R=0:SH=3:Q=0:W=33         :rem 176
120 IFR=L*2THENL=L+1:R=0:GOSUB1000
                                      :rem 45
130 PRINT"{CLR}{ 11 GIU'}"TAB(7)"{RED}LIV
    . :{WHT}"L                        :rem 22
140 PRINT"{ 7 GIU'}"TAB(7)"{GRN}CAV.:
    {WHT}";R+1                        :rem 206
150 FORI=1TO1000:NEXT                 :rem 17
160 PRINT"{CLR}"                      :rem 251
170 IFS>10000ANDQ=0THENPRINT"{HOME}
    { 6 DES}{WHT}NAVE EXTRA":SH=SH+1:Q=1
                                      :rem 176
180 Y1=L+1:X1=L:Y2=L+1:X2=21-L:X3=21-L:Y3
    =22-L:Y4=22-L:X4=L                :rem 201
190 A1=1:A2=1:A3=1:A4=1               :rem 206
200 FORI=0TO21:POKEI+4118,37:POKEI+38422,
    1:POKEI+4580,37:POKEI+3884,1:NEXT
                                      :rem 129
210 FORI=5TO16:POKEI+4206,37:POKEI+38510,
    1:POKEI+4492,37:POKEI+38796,1:NEXT
                                      :rem 195
```

Caverne spaziali

```
220 FORI=1TO22:POKEI*22+4096,37:POKEI*22+
    37888,1:POKEI*22+4117,37:POKEI*22+384
    21,1:NEXT                       :rem 248
230 FORI=5TO18:POKEI*22+4101,37:POKEI*22+
    38405,1:POKEI*22+4112,37:POKEI*22+384
    16,1:NEXT                       :rem 230
240 X=INT(RND(1)*10)+5:Y=INT(RND(0)*6)+8
                                    :rem 236
250 PRINT"{HOME}{ 22 SPAZI}";       :rem 182
260 POKE4096+X+22*Y,W:POKE37888+X+20*Y,1
                                    :rem  29
270 IFX=X1ANDY=Y1THEN860            :rem   6
280 IFX=X2ANDY=Y2THEN860            :rem   9
290 IFX=X3ANDY=Y3THEN860            :rem  12
300 IFX=X4ANDY=Y4THEN860            :rem   6
310 W=W+1:IFW=35THENW=33            :rem 201
320 POKE4096+X1+22*Y1,W+2:POKE37888+X1+22
    *Y1,5                           :rem  65
330 POKE4096+X2+22*Y2,W+2:POKE37888+X2+22
    *Y2,5                           :rem  70
340 POKE4096+X3+22*Y3,W+2:POKE37888+X3+22
    *Y3,5                           :rem  75
350 POKE4096+X4+22*Y4,W+2:POKE37888+X4+22
    *Y4,5                           :rem  80
360 PRINT"{HOME}{ 3 DES}PUNTI";S;:POKE211
    ,16:PRINT"NAVI";SH              :rem 204
370 IF(A1+A2+A3+A4)=0THEN790        :rem   5
380 POKE37154,127:S3=-((PEEK(37152)AND128
    )=0):POKE37154,255:P=PEEK(37137)
                                    :rem  42
390 S1=-((PAND8)=0):S2=((PAND16)=0):S0=((
    PAND4)=0)                       :rem 233
400 XA=S2+S3:YA=S0+S1               :rem 227
410 IFXA=0ANDYA=0THEN620            :rem 203
420 IF(PAND32)=0THEN480             :rem  54
430 POKE4096+X+22*Y,32              :rem 191
440 IFY+YA<1ORY+YA>21THENYA=0       :rem  72
450 IFX+XA<1ORX+XA>21THENXA=0       :rem  68
460 IFPEEK(4096+(X+XA)+(Y+YA)*22)=32THENX
    =X+XA:Y=Y+YA                    :rem 162
470 GOTO620                         :rem 108
480 FORWE=255TO128STEP-5:POKE36877,WE:NEX
    T:POKE36877,0                   :rem  97
490 IFS0THENJ=X:FORI=Y-1TO2STEP-1:GOSUB54
    0:IFBTHENNEXT                   :rem  86
500 IFS1THENJ=X:FORI=Y+1TO21:GOSUB540:IFB
    THENNEXT                        :rem 228
510 IFS2THENI=Y:FORJ=X-1TO1STEP-1:GOSUB54
    0:IFBTHENNEXT                   :rem  80
520 IFS3THENI=Y:FORJ=X+1TO20:GOSUB540:IFB
    THENNEXT                        :rem 231
530 GOTO620                         :rem 105
540 B=-1                            :rem 118
550 IFJ=X1ANDI=Y1THENA1=0:X1=0:Y1=0:S=S+5
    0:B=0                           :rem  33
560 IFJ=X2ANDI=Y2THENA2=0:X2=1:Y2=0:S=S+1
    00:B=0                          :rem  84
570 IFJ=X3ANDI=Y3THENA3=0:X3=2:Y3=0:S=S+1
    50:B=0                          :rem  96
580 IFJ=X4ANDI=Y4THENA4=0:X4=3:Y4=0:S=S+2
    00:B=0                          :rem  99
590 IFPEEK(4096+I*22+J)<>32THENB=0
                                    :rem 170
600 POKE37888+I*22+J,3:POKE4096+I*22+J,39
    :FORK=1TO10:NEXT:POKE4096+I*22+J,32
                                    :rem 147
610 RETURN                          :rem 119
620 IFA1=0THEN660                   :rem 209
630 POKE4096+X1+22*Y1,32            :rem  35
640 X1=X1+(X1>X)-(X1<X)             :rem  31
650 Y1=Y1+(Y1>Y)-(Y1<Y)             :rem  38
660 IFA2=0THEN700                   :rem 209
670 POKE4096+X2+22*Y2,32            :rem  41
680 X2=X2+(X2>X)-(X2<X)             :rem  39
690 Y2=Y2+(Y2>Y)-(Y2<Y)             :rem  46
700 IFA3=0THEN740                   :rem 209
710 POKE4096+X3+22*Y3,32            :rem  38
720 X3=X3+(X3>X)-(X3<X)             :rem  38
730 Y3=Y3+(Y3>Y)-(Y3<Y)             :rem  45
740 IFA4=0THEN780                   :rem 218
750 POKE4096+X4+22*Y4,32            :rem  44
760 X4=X4+(X4>X)-(X4<X)             :rem  46
770 Y4=Y4+(Y4>Y)-(Y4<Y)             :rem  53
780 GOTO260                         :rem 112
790 N=N+1:PRINT"{HOME}{ 6 SPAZI}{WHT}CAVE
    RNA SEGUENTE";                  :rem 147
800 FORG=0TOX:POKE37888+G+22*Y,6:POKE4096
    +G+22*Y,38:NEXT                 :rem  95
810 FORG=X-1TO0STEP-1:POKE37888+G+Y*22,1:
    POKE4096+G+Y*22,33              :rem 213
820 POKE4097+G+Y*22,32:NEXT         :rem  43
830 FORI=128TO255STEP5:POKE36875,I:NEXT:P
    OKE36875,0                      :rem 137
840 PRINT"{CLR}{ 11 GIU'}{ 4 DES}{CYN}BON
    US:{RED}";L*500+R*50:S=S+L*500+R*50:R
    =R+1                            :rem 211
850 FORI=1TO1500:NEXT:GOTO120       :rem  35
860 X1=X-1:X2=X+1:Y1=Y-1:Y2=Y+1     :rem  62
870 FORI=1TO13                      :rem  68
880 POKE4096+X1+22*Y1,38:POKE37888+X1+22*
    Y1,2                            :rem   0
890 POKE36877,128+I*8               :rem 124
900 POKE4096+X2+22*Y1,38:POKE37888+X2+22*
    Y1,2                            :rem 251
910 POKE4096+X1+22*Y2,38:POKE37888+X1+22*
    Y2,2                            :rem 252
920 POKE4096+X2+22*Y2,38:POKE37888+X2+22*
    Y2,2                            :rem 255
930 IFX1>0THENX1=X1-1               :rem 254
940 IFX2<22THENX2=X2+1              :rem  50
950 IFY1>0THENY1=Y1-1               :rem   3
960 IFY2<22THENY2=Y2+1              :rem  55
970 NEXT:POKE36877,0                :rem 179
980 SH=SH-1:IFSH=0THEN1220          :rem  55
990 GOTO120                         :rem 110
1000 T=200+L*60:X=20:Y=11           :rem  78
1010 PRINT"{CLR}{ 5 DES}{GRN}ROUND BONUS"
                                    :rem 228
1020 FORI=4118TO4139:POKEI,37:POKEI+21*22
     ,37:NEXT                       :rem 151
1030 FORI=4118TO4118+22*21STEP22:POKEI,37
     :POKEI+21,37:NEXT              :rem 195
1040 PRINT"{HOME}{ 21 SPAZI}";      :rem 228
1050 FORI=150TO250STEP2:POKE36875,I:NEXT:
     POKE36875,0                    :rem 167
1060 X1=INT(RND(0)*22)              :rem 188
1070 Y1=INT(RND(0)*20)+1:IFPEEK(4096+X1+Y
     1*22)<>32THEN1060              :rem 246
1080 POKE37888+X1+Y1*22,6:POKE4096+X1+Y1*
     22,36:GOTO1100                 :rem  96
1090 IFPEEK(4096+X+Y*22)=36THENS=S+T:GOTO
     1060                           :rem 168
1100 POKE37888+X+Y*22,1:POKE4096+X+Y*22,W
                                    :rem  73
```

```
1110 W=W+1:IFW=35THENW=33           :rem  248
1120 PRINT"{HOME}{PUR}PUNTI:";S;:POKE211,
     13:PRINT"{YEL}TEMPO";T;"{SIN} "
                                    :rem   61
1130 POKE37154,127:S3=-((PEEK(37152)AND12
     8)=0):POKE37154,255:P=PEEK(37137)
                                    :rem   84
1140 S1=-((PAND8)=0):S2=((PAND16)=0):S0=(
     (PAND4)=0)                     :rem   19
1150 POKE4096+X+22*Y,32             :rem  239
1160 IFS0=-1ANDY>2THENY=Y-1         :rem  141
1170 IFS1=1ANDY<21THENY=Y+1         :rem  143
1180 IFS2=-1ANDX>1THENX=X-1         :rem  141
1190 IFS3=1ANDX<20THENX=X+1         :rem  143
1200 T=T-1:IFT=0THENRETURN          :rem  159
1210 GOTO1090                       :rem  199
1220 IFS>H1THENH3=H2:H2=H1:H1=S:GOTO1250
                                    :rem  167
1230 IFS>H2THENH3=H2:H2=S:GOTO1250
                                    :rem   64
1240 IFS>H3THENH3=S                 :rem  156
1250 PRINT"{CLR}{ 13 GIU'}{ 6 DES}{RED}GA
     ME OVER"                       :rem   41
1260 FORI=1TO1500:NEXT:GOTO30       :rem   31
1270 PRINT"{CLR}{ 8 GIU'}{BLK}ATTENDERE P
     REGO..."                       :rem  233
1280 POKE36879,25                   :rem  158
1290 FORI=0TO511:POKE5120+I,PEEK(32768+I)
     :NEXT                          :rem  225
1300 I=5384                         :rem   30
1310 READA:IFA=-1THEN1330           :rem  189
1320 POKEI,A:I=I+1:GOTO1310         :rem   72
1330 POKE36869,205                  :rem  201
1340 RETURN                         :rem  168
1350 DATA 0,8,20,34,73,34,20,8  :rem  179
1360 DATA 0,0,8,20,42,20,8,0    :rem   66
1370 DATA 24,60,106,126,102,60,24,60
                                :rem  218
1380 DATA 24,60,86,126,102,60,24,102
                                :rem  223
1390 DATA 255,219,165,219,219,165,219,255
                                :rem  251
1400 DATA 136,33,0,148,1,72,2,16:rem   19
1410 DATA 0,0,0,24,24,0,0,0     :rem    0
1420 DATA-1                     :rem   63
```



VIC-sound

La nota misteriosa

di **C. Estèves**
trad. ed adatt.
di **M. Cristuib Grizzi, S. Albarelli**
e **M. Anticoli**

**Riconoscere i nomi delle note su spartiti in chiave di violino e di basso è per i principianti un compito difficoltoso.
Perché non farsi aiutare dal computer?**



La cosa più importante da sapere quando si studia o si ha intenzione di studiare musica è quella di riconoscere la posizione delle note su uno spartito in chiave di violino e in chiave di basso.
Il programma qui presentato è stato scritto appositamente per aiutare il musicista principiante a riconoscere istantaneamente il nome delle note sullo spartito, scritto usando le due chiavi più comunemente usate in musica.
Dopo aver fatto partire questo ingegnoso gioco educativo verrannno mostrati due pentagrammi, con relativa chiave, e verrà chiesto al giocatore in quale delle due chiavi musicali vuole cimentarsi.
A tale quesito si può rispondere con "b" per la chiave di basso o con "t" per la chiave di violino o di sol.
Selezionata una delle due chiavi verrà mostrata sul pentagramma corrispondente una nota casuale, della quale bisognerà fornire al computer il nome in notazione anglosassone (seguire l'apposita tabella) entro 10 secondi.
L'indicatore del tempo è mostrato nella parte alta dello schermo, insieme al punteggio corrente.
Se il nome fornito dal giocatore è errato, o il tempo a disposizione è scaduto, il computer proporrà una nuova nota e fornirà altri 10 secondi per trovarne il nome e digitarlo sulla tastiera.
Se, al contrario, il giocatore fornirà la risposta esatta, il computer suonerà la nota indovinata e sommerà al punteggio totale un numero di punti proporzionale al tempo impiegato per rispondere.
Dopo aver proposto 20 note il computer mostrerà il miglior punteggio raggiunto fino a quel momento e darà l'opportunità di ritentare l'impresa.

Note al programma (versione per C16)

Il programma gira sia su C16 che su Plus/4 e utilizza una piccola subroutine in l/m, posta a partire dalla locazione 819, che serve a rilocare il set di caratteri programmati a partire dalla locazione 15360.
Se si desiderasse analizzare il listato per scovare eventuali errori di battitura, è necessario, per vederlo normalmente, premere contemporaneamente i tasti RUN-STOP e RESET (quest'ultimo si trova sul fianco destro del computer).
Facendo ciò si entrerà nella subroutine di sistema MONITOR: per uscirne e analizzare il listato è necessario digitare "X" e premere RETURN.
Dopodiché basterà un semplice LIST, e subito dopo un HELP (F7), per vedere in che linea si trova e qual è l'errore di battitura che ha bloccato il programma.
Per facilitare la digitazione del listato può essere utile programmare i tasti di funzione mediante l'istruzione KEY (vedere il libretto allegato al computer per eventuali informazioni sul comando).
Le parole più utili da inserire nei tasti funzione possono essere:

F1 "DATA"
F3 "POKE"

F5 "PRINT"
F2 "FOR"
F4 "THEN"
F6 "GOSUB"
F8 "GOTO"

È consigliabile non riprogrammare il tasto "F7", perché contiene la funzione "HELP", utile per correggere gli errori di battitura.

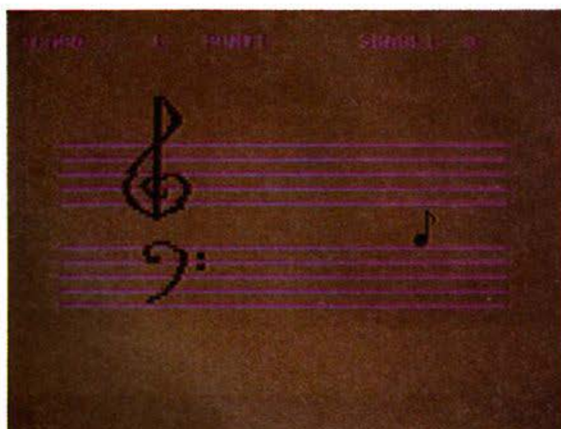
Tabella di conversione dei nomi delle note.

C = DO
D = RE
E = MI
F = FA
G = SOL
A = LA
B = SI

La nota misteriosa
versione per C64

```
100 DIM LOW%(25),HI%(25)              :rem  212
110 PRINT"{BLK}{CLR}{ 13 GIU'}{ 12 DES}AT
    TENDERE..."                       :rem  229
120 POKE53281,12:POKE53280,12         :rem   78
130 REM ****ROUTINE DI INIZIALIZZAZIONE**
    **                                :rem   78
140 NO$="EFGABCDEFGABCDEFGABCDEFGA"
                                      :rem  126
150 FORT=1TO25:READ LOW%(T):NEXT:FOR T=1T
    O25:READHI%(T):NEXT               :rem  235
160 FORT=0TO63:READD:POKE704+T,D:NEXT:POK
    E2040,11                          :rem   14
170 FORT=832TO1020:READD:POKET,D:NEXT:POK
    E2041,13:POKE2042,14              :rem  213
180 POKE2043,15                       :rem   35
190 V=53248:POKEV+21,0                :rem  220
200 POKEV,255:POKEV+16,0:POKEV+39,0:POKE5
    3280,12:POKE53281,12              :rem   51
210 POKEV+2,90:POKEV+3,84:POKEV+4,88:POKE
    V+5,126:POKEV+23,14:POKEV+29,14
                                      :rem   35
220 POKEV+6,90:POKEV+7,163:POKEV+40,0:POK
    EV+41,0:POKEV+42,0                :rem  247
230 S=54272:FORT=STOS+24:POKET,0:NEXT
                                      :rem   71
240 POKES+24,13:POKES+5,17:POKES+6,243
                                      :rem  203
250 PRINT"{CLR}{ 10 GIU'}{ 13 DES}LA NOTA
     MISTERIOSA":FORT=6TO20:POKES,LOW%(T)
                                      :rem  110
260 POKES+1,HI%(T):POKES+4,17:FORT1=1TO15
    0:NEXT:POKES+4,16                 :rem   32
270 FORT2=1TO40:NEXT:NEXT             :rem  109
280 CH$="":PRINT"{ 2 GIU'}{ 4 DES}VUOI:
    {RVS}A{OFF}LTI, {RVS}B{OFF}ASSI O
    {RVS}T{OFF}UTTO":INPUT "{ 4 DES}";CH$
                                      :rem  247
290 IFCH$<>"A"ANDCH$<>"B"ANDCH$<>"T"THEN
    PRINT"{ 5 SU}" :GOTO280           :rem  208
300 LW=-(CH$="B"ORCH$="T")-(CH$="A")*13
                                      :rem    5
310 TP=-(CH$="B")*13-(CH$="A"ORCH$="T")*2
    5                                 :rem  152
320 POKE646,6:PRINT"{CLR}{ 6 GIU'}"
                                      :rem  202
330 LI$="{ 34 *}":FORT=1TO5:PRINTSPC(3)LI
    $:NEXT                            :rem  110
340 PRINT"{GIU'}"                     :rem  121
350 FORT=1TO5:PRINTSPC(3)LI$:NEXT
                                      :rem  124
360 PRINT"{HOME}TEMPO :{ 7 DES}PUNTI
    { 7 DES}SBAGLI"                   :rem   20
370 FORTR=1TO20                       :rem  154
380 NN=INT(RND(0)*(TP-LW)+LW):IFNN=NO THE
    N380                              :rem  219
390 POKES,LOW%(NN):POKES+1,HI%(NN)
                                      :rem   53
400 IFNN=1ORNN=25ORNN=13THENPOKE752,127:P
    OKE754,240:GOTO420                :rem  130
410 POKE752,1:POKE754,0               :rem   36
420 POKEV+1,(194-4*NN)+8*(NN>13):POKEV+21
    ,15                               :rem  180
430 TS=10                             :rem  220
440 FORT=0TO500:NEXT:TS=TS-1:PRINT"{HOME}
    { 9 DES}";TS:GETG$                :rem  158
450 IFG$<>""THENPOKE1934,(ASC(G$)-64)AND2
    55                                :rem  150
460 IF(G$<>""ANDG$<>MID$(NO$,NN,1))ORTS=0
    THENWR=WR+1:POKE198,0             :rem  198
470 PRINT"{HOME}{ 33 DES}";WR         :rem   32
480 IFG$<>MID$(NO$,NN,1)ANDTS<>0THEN440
                                      :rem   48
485 IFTS<>0THENPOKES+4,17:FORT=1TO300:NEX
    T:POKES+4,16                      :rem  101
490 SC=SC+TS:PRINT"{HOME}{ 20 DES}";SC;
                                      :rem   70
500 FORT=0TO1500:NEXT                 :rem   31
510 NO=NN:NEXT                        :rem  133
520 POKEV+21,0:PRINT"{CLR}{ 11 DES}
    { 7 GIU'}PUNTEGGIO:";SC           :rem   23
530 PRINT"{ 3 GIU'}{ 9 DES}HAI SBAGLIATO"
    ;WR;"VOLTA(E)"                    :rem  136
540 PRINT"{ 3 GIU'}{ 6 DES}VUOI GIOCARE A
    NCORA?"                           :rem  123
550 GETA$:IFA$=""THEN550              :rem   87
560 IFA$<>"S"ANDA$<>"N"THEN550        :rem  173
570 IFA$="S"THENRUN                   :rem  136
580 REM ****DATI NOTE (BYTES BASSI)****
                                      :rem  115
590 DATA 71,152,71,12,233,97,104,143,48,1
    43,24,210                         :rem  153
600 DATA 195,209,31,96,30,49,165,135,162,
    62,193,60,99                      :rem   72
610 REM ****DATI NOTE (BYTES ALTI)****
                                      :rem   37
620 DATA 5,5,6,7,7,8,9,10,11,12,14,15
                                      :rem    6
630 DATA 16,18,21,22,25,28,31,33,37,42,44
    ,50,56                            :rem  253
640 REM ****DATI PER GLI SPRITE****
                                      :rem  138
650 DATA 000,002,000,000,003,000,000,002
                                      :rem  112
660 DATA 128,000,002,064,000,002,032,000
                                      :rem  136
670 DATA 002,032,000,002,032,000,002,064
                                      :rem  133
680 DATA 000,002,000,000,002,000,000,002
                                      :rem  114
690 DATA 000,000,002,000,000,002,000,000
                                      :rem  113
700 DATA 122,000,000,254,000,001,254,000
                                      :rem  129
710 DATA 001,254,000,001,254,000,000,252
                                      :rem  135
720 DATA 000,000,120,000,000,000,000,000
                                      :rem  106
730 DATA 0,224,0,0,208,0,0,216,0,0,204,0,
    0                                 :rem  213
740 DATA 206,0,0,199,0,0,199,0,0,199,0,0,
    206,0                             :rem  186
750 DATA 0,204,0,0,216,0,0,240,0,0,224,0,
    1                                 :rem  212
760 DATA 192,0,3,192,0,6,192,0,12,192,0,2
    4,192                             :rem  197
```

```
770 DATA 0,48,192,0,96,192,0,224,192,0,25
    5                              :rem   19
780 DATA 48,111,128,97,248,192,195,96,96,
    198,96,48,195,104,48,193      :rem  200
790 DATA 232,48,96,248,96          :rem  233
800 DATA 112,96,224,56,96,192,28,99,192,7
                                   :rem  254
810 DATA 111,0,1,248,0,0,96,0,0,0,0,0,0,0
    ,0,0,0                         :rem  180
820 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                   :rem   72
830 DATA 0,248,0,3,6,0,6,3,128,6,0,198,3,
    192                            :rem   99
840 DATA 198,3,192,192,0,0,198,0,1,134,0,
    1                              :rem    3
850 DATA 128,0,3,0,0,3,0,0,6,0,0,12,0,0
                                   :rem   61
860 DATA 24,0,0,112,0,1,192,0,3,0,0,0,0,0
    ,0,0,0                         :rem  178
870 DATA 0,0,0,0,0,0               :rem  181
```

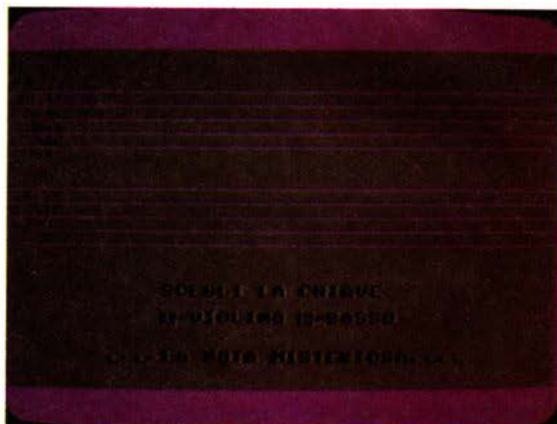


La nota misteriosa
versione per VIC 20

```
10 FORI=1TO210:READA:X=X+A:NEXT:IFX<>2886
   0THENPRINT"ERRORE NEI 'DATA'":END
20 POKE56,60:POKE55,0:CLR:VOL8
30 FORI=819TO848:READA:POKEI,A:NEXT
40 DATA160,0,185,0,208,153,0,60,185,0,209
   ,153,0,61
50 DATA185,0,210,153,0,62,185,0,211,153,0
   ,63,136,208,229,96
60 PRINT"{CLR}"CHR$(142):COLOR0,4,1:COLOR
   4,4
70 F$="{ 39 C}"
80 C$=CHR$(13):PRINT"{HOME}{ 9 GIU'}{WHT}
   C"F$F$"{ 12 C}LA NOTA MISTERIOSA
   { 13 C}"F$F$
90 FORI=1TO3000:NEXT
200 COLOR0,7:COLOR4,7:PRINTCHR$(147):PRIN
    T"{BLU}{ 10 GIU'}"TAB(14)"ATTENDERE P
    REGO":SYS819
210 FORI=0TO7:POKE15360+I,PEEK(16056+I):N
    EXT:FORI=0TO7:READX:POKE15576+I,X:NEX
    T
220 DATA0,0,0,0,0,60,66,66,66,66,60,0,0,0
    ,0,255,0,60,66,66,66,66,60,255
230 DATA60,126,227,193,129,129,129,255,12
    9,131,135,142,156,184,240,255,224,192
240 DATA128,128,128,128,128,255,128,240,2
    48,140,134,131,129,255,129,129,129,13
    0,132,136
250 DATA144,255,128,128,128,128,128,128,0
    ,0,0,0,0,32,112,112,97,62,128,128,64,
    32,16,8,4
260 DATA255,96,195,135,136,136,132,128,25
    5,1,7,12,24,48,96,96,255
270 DATA0,0,0,24,24,0,0,255,240,56,12,6,3
    ,131,195,255,195,131,3,3
280 DATA3,3,3,255,3,3,6,12,24,48,96,255,3
    3,30,0,0,0,0,0,255,1,2,4,8,8,5,2,255,
    3,1,0,0,0
290 DATA0,0,255
300 FORI=0TO7:READX:POKE15584+I,X:POKE155
    92+I,PEEK(16160+I):NEXT
310 FORI=15600TO15615:READX:POKEI,X:NEXT:
    FORI=15624TO15743:READX:POKEI,X:NEXT
320 FORI=15824TO15831:READX:POKEI,X:NEXT
330 COLOR0,2:COLOR4,5:POKE65298,PEEK(6529
    8)AND251
340 POKE65299,PEEK(65299)AND3OR60
350 DATA7,118,169,262,345,383,453,516,571
    ,596,643
360 DATA685,704,739,770,798,810,834,854,8
    64,881,897
370 PRINTCHR$(147):FORI=3152TO3351:POKEI,
    29:NEXT
380 FORI=3432TO3631:POKEI,29:NEXT
390 POKE3155,31:POKE3195,33:POKE3235,34:P
    OKE3275,35:POKE3315,36:POKE3355,37
400 POKE3354,38:POKE3314,39:POKE3274,40:P
    OKE3234,41:POKE3475,42
410 POKE3515,42:POKE3474,43:POKE3514,44:P
    OKE3554,45:POKE3593,46:POKE3473,47
420 PRINT"{HOME}{ 22 GIU'}{GRN}"TAB(7)"&
    {PUR}&{BLK}&{BLU}&{BLK}LA NOTA MISTER
    IOSA{BLK}&{PUR}&{GRN}&{CYN}&{HOME}"
430 PRINT"{ 16 GIU'}"TAB(11)"{BLK}SCEGLI
    LA CHIAVE{GIU'}"
440 PRINTTAB(11)"{RVS}T{OFF}=VIOLINO
    {RVS}B{OFF}=BASSO":GOSUB770
450 R=0:DIMP(18):DIMN(18):FORI=0TO18STEP2
    :P(I)=3079+R:R=R+40:NEXT
460 DIMN1(24):FORN=24TO3STEP-1:READN1(N):
    NEXTN
470 N(0)=68:FORH=1TO18:N(H)=(N(H-1)-1)+(7
    *-(N(H-1)=65)):NEXT
480 FORI=3761TO3782:POKEI,32:POKEI+80,32:
    NEXT
490 GOSUB740:C=5:SC=0:Q=L*40:FORI=2048TO2
    846:POKEI,0:NEXT
500 FORI=0TO30STEP10
510 J=INT(RND(X)*13)+3:IFJ=OJOR(L=7ANDJ>1
    2)THEN510
520 OJ=J:J1=J:E=29:E1=30:E2=27:E3=28:PRIN
    T"{HOME}"TAB(5)"{RVS}{RED}PUNTI{OFF}
    {BLK}"SC
530 PRINT"{HOME}"TAB(26)"{RVS}{PUR} TEMPO
     {OFF}{BLK}"9:IFJ1=INT(J1/2)*2THENPOK
    EP(J1)+I+Q,E1:GOTO550
540 POKEP(J1-1)+I+Q,E2:POKEP(J1+1)+I+Q,E3
    :POKEP(J1-1)-1+I+Q,E:POKEP(J1-1)+1+I+
    Q,E
550 IFE1=29ORE1=32THEN660
560 TI$="000000":POKE239,0
570 GETN$:IFN$=""THENT=TI:IFT<600THENPRIN
    T"{HOME}"TAB(33)9-INT(T/60):GOTO570
580 IFN$=""THEN610
590 IFASC(N$)<65THENN$="":GOTO610
600 POKE3759+I,ASC(N$)-64:IFASC(N$)=N(J1+
    Z)THEN710
```

VIC-sound

```
610 FORT=0TO500:NEXTT:IFJ1=3THENE=32:E1=3
    2:E2=32:E3=29
620 IFJ1=13THENE1=32:E2=29:E3=32
630 IFJ1=14ORJ1=15THENE=32:E1=32:E2=32:E3
    =32
640 IFJ1>3ANDJ1<13THENE1=29:E2=29:E3=29
650 POKE3759+I,32:GOTO530
660 NEXTI
670 C=C-1:IFC>0THEN500
680 IFSC>SITHENSI=SC
690 PRINT"{HOME}"TAB(5)"{RVS}{RED}PUNTI
    {OFF}{BLK}"SC:PRINTTAB(16)"{RVS}[<7>]
    REC {OFF}{BLK}"SI
700 PRINT"{HOME}{ 19 GIU'}"TAB(10)"{RVS}T
    {OFF}=VIOLINO {RVS}B{OFF}=BASSO":GOSU
    B770:GOTO480
710 POKE2737+I,0:SC=SC+(600-T)
720 IFJ1>2ANDJ1<16THENSOUND1,N1(J1+X1+L),
    25
730 GOTO610
740 F=15+L:G=8+L:K=-1:GOSUB760
750 F=8+L:G=15+L:K=1:GOSUB760:RETURN
760 FORP=FTOGSTEPK:SOUND1,N1(P),20:NEXTP:
    RETURN
770 POKE239,0:GETKEYA$:IFA$="T"THENL=0:Z=
    0:X1=0:GOSUB800:RETURN
780 IFA$="B"THENL=7:Z=-2:X1=5:GOSUB800:RE
    TURN
790 POKE65298,196:POKE65299,208:PRINTCHR$
    (147):END
800 FORI=3072TO3151:POKEI,32:NEXT:RETURN
```



PRINT

Recensioni Libri

LOGO: POTENZA E SEMPLICITÀ
di **E. Baldi** e **M. Divizio**
Gruppo Editoriale Jackson



Siamo in piena era informatica e tutti, bene o male, desiderano imparare a programmare i calcolatori. Tuttavia, chi già non disponga di una certa cultura matematica e di una mentalità logica incontra una certa difficoltà iniziale nell'imparare la programmazione con un linguaggio come il BASIC. Esiste invece un linguaggio di programmazione appositamente dedicato a chi si avvicina per la prima volta ai computer, il LOGO. Il LOGO, largamente impiegato per sviluppare una mentalità logica nei bambini, sfrutta la tecnica dei grafici "a tartaruga": un simbolo – la tartaruga – che spostandosi sullo schermo in alta risoluzione lascia una traccia che può essere impiegata per creare disegni di qualsiasi forma.
Questo libro sviluppa esempi e tecniche della programmazione in LOGO, mettendo subito in grado il bambino o l'adulto di avere un approccio immediato con il computer e di sviluppare le proprie doti logiche e razionali.
Si analizzano le istruzioni di questo linguaggio, le procedure e le sottoprocedure e tutte le tecniche di programmazione, come i cicli, i test, la ricorsività e le operazioni sui numeri. Molto spazio viene dedicato ai grafici in alta risoluzione ed a come ottenere cerchi, ellissi, parabole ed archi.
Un libro impostato sulla didattica, che sarà senz'altro utile a quei genitori o insegnanti che desiderano creare un facile e simpatico approccio del bambino al computer.

LINGUAGGIO MACCHINA DEL COMMODORE 64
di **D. Lawrence** e **M. England**
Gruppo Editoriale Jackson

La programmazione in linguaggio macchina affascina sempre più i possessori di home computer che hanno già acquisito una approfondita conoscenza del BASIC.
Uno dei principali "handicap" incontrati fino a poco fa da chi volesse acquisire informazioni circa la programmazione in linguaggio macchina del C64 era la quasi totale assenza di letteratura in materia, volendo prescindere dai testi tecnici in lingua inglese, non certo alla portata di tutti.
Questo libro, traduzione di un testo molto diffuso nei paesi anglofoni, offre forse il miglior approccio per l'apprendimento del linguaggio macchina: esso permette infatti di entrare nel linguaggio ASSEMBLY attraverso la programmazione in BASIC. Esaminando i vari "moduli" di programma BASIC, che vengono via via presentati ed analizzati dal libro, si acquisiscono senza sforzo tutte quelle nozioni che normalmente fanno sudare – per la loro difficoltà iniziale di comprensione – le classiche sette camicie... Alla fine della lettura del libro ci si troverà a disporre di un ottimo programma editor/assembler, che attenderà soltanto di essere sfruttato per i primi esperimenti di programmazione "autonoma" in linguaggio macchina. L'ditor/assembler, scritto completamente in BASIC, viene "costruito" collegando l'uno con l'altro i vari moduli contenuti nel libro ed offre tutti i vantaggi di programmazione in l/m propri dell'editor rispetto ad un comune assemblatore/disassemblatore.



I libri recensiti possono essere ordinati utilizzando il coupon a pag. 82

7 Note Bit

Oggi la musica si impara così.

SPEDIZIONE IN ABBONAMENTO POSTALE GRUPPO II/70
L. 10.000
NOTE 7BIT
IMPARARE LA MUSICA COL C64

C64

LA MUSICA E... SUBITO ALLA TASTIERA CON IL C64

È in edicola a L. 10.000
Questa favolosa tastiera compresa nel prezzo

per C64

Aut. D.M. 4/275256 del 2/3/85

Il 1985 è l'anno mondiale della musica... preparati con 7 note Bit, la nuova, fantastica opera che in soli 15 fascicoli quattordicinali (ciascuno corredato da una cassetta software per Commodore 64) ti insegna veramente la musica. E puoi suonare subito

Non perdere il primo numero. Pensa, compresa nel prezzo c'è anche la stupenda tastiera professionale per il tuo Commodore 64.



GRUPPO EDITORIALE JACKSON
San Francisco-Londra-Milano

IN COLLABORAZIONE CON SIEL

GRUPPO ETHOS

....E VINCI 30 STRAORDINARI COMMODORE PLUS/4



input·output

Piccoli annunci

Per C64 vendo cassetta con i seguenti 10 giochi: Roulette, Torri di Hanoi, Death, Painter, Lo Scalatore Folle, I Due Serpenti, Trenchfire, Poker-Indiano, Il Gioco del 15, Tombola a sole L. 12.000 più spese postali. Tutti i programmi sono registrati col Turbo Tape, compreso nella cassetta. Roberto Caputo - Via Roma, 17 - 81030 Teverola (CE) - Tel. 081/8118104

Per VIC 20 vendo circa 100 programmi in blocco a L. 50.000 più L. 2.000 per spese di spedizione. Solo listati. Fabio Ferrari - Via Araldi, 5 - 29100 Piacenza - Tel. 0523/753283

Vendo VIC 20 seminuovo più registratore, 5 cartucce, cassette. Affarissimo. Fabrizio Crenca - Via Isidoro Croce, 12 - 00046 Grottaferrata (RM) - Tel. 9459086

Vendo la traduzione delle istruzioni del videogioco "Solo Flight", "Simulazione di Volo" 1 a L. 5.000. Anna Budal - Località Log. 192 - 34100 S. Dorligo della Valle (TS) - Tel. 040/813145

Per VIC 20 vendo super espansione 3 Kbyte più libro di grafica a L. 60.000. Espansione 3-16 Kbyte L. 110.000 più L. 2.000 per spese di spedizione. Fabio Ferrari - Via Araldi, 5 - 29100 Piacenza - Tel. 0523/753283

Per VIC 20 vendo cassette con programmi in LM di Program, Playgame, Load'n'Run, Arcade, in totale 9, a L. 50.000. Fabio Ferrari - Via Araldi, 5 - 29100 Piacenza - Tel. 0523/753283

Vendo o scambio programmi di giochi, utility per CBM 64. Invio lista programmi a chi ne fa richiesta. Scrivere o telefonare dalle 13 alle 14. Pietro Ghisellini - Via Pavone, 15 - 44100 Ferrara - Tel. 0532/21718

Vendo VIC 20 usato poco, più cassetta contenente 75 giochi, di cui molti in LM a L. 150.000 non trattabili. Telefonare ore pasti e chiedere di Paolo. Paolo Caronti - Via Dei Bognetti, 9 - 20141 Milano - Tel. 8438958

Vendesi programmi su cassetta per VIC 20 inespanso. Giochi con ottima grafica, utility, prezzi stracciati! Chiedo e offro la massima serietà. Richiedere lista gratuita. Stefano Cesana - Via Isonzo, 13 - 20042 Albiate (MI)

Vendo CBM in perfetto stato, alimentatore, registratore dedicato, joystick, manuale italiano e due splendidi libri sul CBM 64. Inoltre più di 40 giochi (Arcadia, Pit Stop, Krazy Kong, Soccer, Scacchi); più di 30 riviste e un utilissimo apparecchio per utilizzare qualsiasi tipo di registratore. Tutto a L. 680.000 non trattabili. Solo zona Torino e provincia. Telefonare ore 10-12. Eugenio Santomarco - Via Gazzera, 53/B - 10073 Ciriè (TO) - Tel. 011/3097052

Vendo VIC 20 con registratore come nuovi, imballati. Massima serietà. Gerardo Izzo - Via Bellini, 1 - 81042 Calvi Risorta (CE) - Tel. 0823/87426

Eccezionale! Vendo, cambio programmi per Commodore 64 (Pole Position, International Soccer, Scacchi e tanti altri) a L. 2.000 l'uno, Sintetizzatore di voce Sam Reciter, trasformatore video da 40 a 80 colonne. Telefonare o scrivere. Alfredo Benincasa - Via Buccini, 4 - 81100 Caserta - Tel. 0823/324778

Per CBM 64 dispongo di formidabili utility come: Clone 2, K.P. Pirata, Mimic 5, di copiatore e riparatore di singola traccia; gestionali vari, ingegneria, word processor e gli ultimi giochi '84. Paola Sbarzella - Via S. Martino, 264 - 05100 Terni - Tel. 0744/56870

Cerco giochi per C64 in cambio di altri bei giochi: Pitfall 2, Zaxxon, Dallas, Frogger, H.E.R.O. e molti altri. Davide Lopez Nunes - Via G. Frua, 21/10 - 20146 Milano - Tel. 02/486623

Compro, vendo, cambio circa 150 tra giochi e programmi di utilità su cassette per C64. Scrivetemi per ricevere la lista oppure mandatemi la vostra. Carmine Collaro - Vico I Comizio - 80041 Boscoreale (NA) - Tel. 8582159

Vendo cartucce per Atari: Popeye, QBert, Junge Hunt, Super Breakout, Phoenix e altre a L. 25.000 cadauna. Scrivere o telefonare a: Flavio Tiberi - Via S. Balbina, 1 - 00010 S.Polo (RM) - 560114

Vendo per C64 cassetta con i seguenti giochi: Calcio, Shuttle, Galaxi, Dig Dug, One On One, Atzec, Super Pipeline, Survivor, tutti registrati con il Turbo Tape a L. 15.000, oppure distintamente a L. 2.000 cadauno. Sergio Molino - Via Annunziata, 117 - 12064 La Morra (CN) - Tel. 0173/50348

Vendo consolle Atari in ottime condizioni con 5 cassette (Pac-Man, Dodgèem, Combat, Pelès Soccer, Maze Craze) a sole L. 250.000. Chiamare: Luigi Cardone - Corso Vittorio Emanuele, 41/A - 80059 Torre del Greco (NA) - Tel. 081/8817182

Per VIC 20 vendo programmi bellissimi con e senza espansione a prezzi bassissimi. Tutti registrati su cassetta. Inoltre vendo VIC 20 con 5 mesi di vita più espansione 16 Kbyte, "Introduzione al BASIC" Vol. 1, joystick, 2 cartucce e tanti programmi al prezzo di L. 378.000. Carlo Marella - Via M. del Carmine, 57 - 03023 Ceccano (FR) - Tel. 600363

Riparo piccoli sistemi Commodore VIC 20 - CBM 64 e loro periferiche. Fiori Ettore - Via Fra Bartolomeo, 4 - 20146 Milano - Tel. 02/471521

Vendo VIC 20, Datassette C2N, libro "Alla scoperta del VIC 20", cartridge "Gorf", il tutto a L. 250.000. Marco Dagradi - Via Saponaro, 20 - 20142 Milano - Tel. 02/8263925

Cercansi programmi gestionali per Commodore 64. Per informazioni telefonare a: Loris Zucchetti - Corso Milano, 4 - 25056 Pontedilegno (BS) - Tel. 0364/91018

Vendo per C64 cassetta con 6 fantastici giochi registrati con Turbo Tape, compreso nella cassetta, a L. 30.000 più spese postali: Baseball, Frogger, Congo Bongo, Neptun 7801, Pac Man, Seawolf. Scrivete a: Marco Tofanelli - Via Aurelia, 89 - 58016 Orbetello Scalo (GR)

Ho molti programmi, sia games che utility. Cambio: Skramble, Baseball, Basket, Pipeline, Fort Apocalypse, Pogo Joe, Monster Attack, Popeye, Atzec Challenge, e tanti altri. Canbio programmi per fatture, Totocalcio, Magic Desk. Tutti su cassetta. Filippo Morri - Via Olivella, 5 - 47031 Serravalle di Repubblica di San Marino - Tel. 900245

Fantastico! Per CBM 64 vendo o cambio il miglior software disponibile sul mercato a prezzi stracciatissimi. Programmi gestionali, didattici e games superbi. Richiedere la lista scrivendo o telefonando a: Alfonso Mansi - Via F. Provenzale, 49 - 80141 Napoli - Tel. 081/7802166

Vendo a prezzi ottimi giochi per CBM 64. Esempio: Donkey Kong L. 7.000, Jungle Hunt L. 7.000. Congo Bongo L. 5.000, ecc. su cassette. Invio lista gratis. Guido Nocciolini - Via P. Nenni, 6 - 53100 Siena - Tel. 0577/285864

Vendo per C64 4 giochi Arcade su Turbo Tape a L. 30.000: Baseball, Star Trek, Hero, Zaxxon. Tutti originali. Richiedere lista. Roberto Costagliola - Via Dardanelli, 17 - 00195 Roma - Tel. 06/314909

Vendo Commodore 64 con drive e stampante 801, 1.000 programmi utilità e giochi bellissimi. Prezzo basso. Giuseppe Borracci - Via Mameli, 15 - 33100 Udine - Tel. 0432/291665

Cerco amici alessandrini possessori di C64 per scambio programmi e magari per fondare un club (zona Alessandria). Claudio Veronese - Via Tonso, 75 - 15100 Alessandria

Vendo cassetta con molti programmi (giocosi) per VIC 20 a L. 30.000. Affarone, scrivetemi o telefonatemi, rispondo a tutti fino ad esaurimento merce. Gianluca Maranzana - Via A. De Zigno, 8 - 35100 Padova - Tel. 049/27694

Vendo, cambio programmi per C64: Miner, Burger Time, Mr. Robot. Dieci giochi L. 20.000, Turbo Tape L. 10.000; con 30 giochi Turbo Tape gratis. Per la lista completa scrivetemi. Massima serietà. Giancarlo Turco - Via Ferrovia - 81057 Teano (CE) - Tel. 0823/875928

Cambio, vendo software per il grande VIC 20, con oltre 200 programmi (giochi, utility, ecc.). Invio lista gratis. Massima serietà. Enrico Rizza - Via Vasari, 42 - 66011 Bucchianico (CH) - Tel. 0871/685319

Vendo cassetta con 15 giochi in LM per il VIC 20 inespanso a L. 15.000. Telefonare tutti i giorni dalle 14 alle 18.30. Eugenio Ingoglia - Contrada Marcato - 91028 Partanna (TP) - Tel. 0924/88053

Vendo a prezzo favorevole VIC 20 nuovo più 3 libri: "Introduzione al BASIC" con due cassette, "Impariamo a programmare in BASIC col VIC e CBM", "Alla scoperta del VIC" più 3 giochi in 2 cassette, 3 cassette di "Conoscere il computer direttamente da computer". Per accordi scrivere a: Francesco Florio - Via Palazzo, 126 - 98019 Torre Faro (ME) - Tel. 090/393713

Cambio programmi per CBM 64 (ne ho 700). Inviare liste e proposte. Rispondo a tutti. Scrivere a: Fabrizio Raucci - Via Terme di Traiano, 38 - 00053 Civitavecchia (RM)

Vendo, cambio programmi per CBM 64. Posseggo oltre 500 titoli tra cui le ultimissime novità del mercato. Vendo sia su nastro (L. 3.000 a programma) che su disco (L. 3.000 per programmi Uniload, L. 5.000 per quelli Multiload). Chi è interessato può telefonare ad Angelo dalle 15.30 alle 19.30, tranne sabato e domenica, allo 081/927965

Vendo VIC 20, registratore più 10 cassette in LM, 2 libri sul VIC 20, giochi in LM e BASIC, joystick: il tutto poco usato e registratore in garanzia a L. 400.000. In regalo libro con programmi BASIC. Si prega di telefonare dopo le ore 20.15. Enrico Dal Pezzo - Viale Einaudi, 15/i - 20020 Arese (MI) - Tel. 02/9382067

Vendo per VIC 20 programmi in versione base, a 8 e 16 Kbyte, anche tratti da cartridge. Telefonare ore serali o scrivere per lista. Gianni Gaudino - Via Graglia, 18 - 10136 Torino - Tel. 011/352830

Vendo Atari 2600 più 9 giochi Atari fra cui Tank, Space Invaders e Il Bruco a sole L. 180.000 o cambio con 32 Kbyte RAM per VIC 20. Vendo cassette Atari a poco (da L. 25.000 a L. 50.000) Phones, ecc. Stefano Chelli - Via Lorenzo Quartieri - 55042 Forte dei Marmi (LU)

Vendo programmi su nastro. Ne ho circa 300 tra cui Breakdance, H.E.R.O., Pit Fall 1/2, Beamrider, Jungle Hunt, Ms. Pacman, Zaxxon, Summer Games, Decathlon, Popeye, Caesar, The Cat, ecc.). Annuncio sempre valido. Maurizio Borrelli - Via Firenze, 32 - 81100 Napoli - Tel. 081/281672

Vendo, compro, cambio programmi per C64. Inviare la vostra lista con prezzi a: Giovanni Marino - Via Ragnana, 28 - 57012 Castiglioncello (LI) - Tel. 0856/753017

input·output

Piccoli annunci

Compro, vendo, cambio giochi e programmi di vario genere per Commodore 64. Ho più di 100 giochi, ma vorrei centuplicarli. Inviate le vostre liste. Annuncio sempre valido. Eno Manica - Via S. Croce, 49 - 88074 Crotone (CZ)

Vendo, compro, cambio programmi su cassetta per C64. Ne posseggo circa 200, in vendita da L. 600 a L. 2.000 l'uno, utility comprese. Adriano Bochese - Via G. Marconi, 8 - 37050 Belfiore d'Adige (VR)

Vendesi per C64 combinatore telefonico comprendente interfaccia elettronica e software. L'interfaccia va collegata alla presa di un normale telefono; il software, su nastro o disco, comprende una subroutine per la combinazione automatica del numero ed un programma di rubrica con stampa su carta dei nominativi. Prezzo L. 55.000. Tutto compreso. Roberto Allario - Via Migliarone, 3 - 10091 Alpignano (TO) - Tel. 011/9672206

Compro, cambio, vendo software per CBM 64 nella sola zona di Catania. Fabio Bellassai - Via Fratelli Bandiera, 13 - 95100 Catania - Tel. 095/415353

Vendo, cambio programmi su cassetta per C64 (circa 150). Pole Position, Zaxxon, BC, Frogger, Popeye, Decathlon, Simons' BASIC, Magic Desk, ecc. tutti in LM a prezzi bassi (L. 3.000 - 5.000 - 7.000). Due programmi in regalo a chi invia la sua lista (per scambio o acquisto); scrivere a: Roberto Ianniello - Via Luciano, 60 - 80078 Pozzuoli (NA)

Compro, vendo, cambio software per VIC 20 inespanso, specialmente programmi utility e giochi. Maurizio Ruozi - Via Grazia Deledda - 42020 Albinea (RE) - Tel. 0522/59604

Eccezionale! Vendo per L. 50.000 una cassetta di 10 videogame per CBM 64 (Fort Apocalypse, Calcio, Juice, Pit Stop, Kong, Frogger, AMC, Biliardo, Ski, F104) più Turbo (velocizzatore che trova, salva e verifica circa 10 volte più velocemente i vostri programmi). Scrivere o telefonare ore pasti. Alberto Busiello - Via Alba, 15 - 12038 Savigliano (CN) - Tel. 0172/2966

Per C64 vendo: Football, Basket, Pole Position, Sintetizzatore, Totocalcio e tantissimi altri a sole L. 5.000 cadauno più spese. Giovanni Pugliese - Via A. Volta, 93 - 74100 Taranto - Tel. 099/413769

Vendiamo e compriamo programmi per CBM 64 e VIC 20 su disco e cassetta. Esempi: Pole Position, Zaxxon, Dig Dug e tanti altri a prezzi stracciati. Massima serietà. Marcello Ghilardi - Via E. Fermi Palazzo Morrone - 84012 Angri (SA)

Cambio, vendo programmi per CBM 64, ZX Spectrum 16-48 Kbyte e VIC 20. Assicuro prezzi convenientissimi. Inviare L. 300 per il listato. Andrea Vicari - Via Voltone, 72 - 47031 Repubblica di San Marino - Tel. 0541/991716

Vendo fantastico Commodore VIC 20 con o senza: registratore, espansione RAM 3-8-16 Kbyte, cartuccia Turbo Tape e anche sfuse cartucce Radar Rat Race, Protector, Aggressor, Raid On Fort Knox. Angelo Preatoni - Via Aurelia, 198 - 17023 Ceriale (SV) - Tel. 0182/90346

Cerco Nr. 1 e 3 di "Supervic" con cassetta programmi per C64. Vendo VIC 20 più 1 libro e cassette programmi a L. 80.000. Corrado Rosati - Via A. Spunticchia, 14 - 00018 Palombara Sabina (RM) - Tel. 0774/66437

Per VIC 20 vendo giochi in cassetta e cartridge e il libro "Giochi - giochi per il VIC 20". Invio lista gratis. Scrivere a: Michele Sirtori - Via Filangieri, 5 - 35100 Padova

Scambio, vendo giochi per VIC 20. Richiedete il listino dei giochi. Andrea Francavilla - Via C. Battisti 3 Strada, 6 - 64014 Martinsicuro (TE)

Vendo, cambio per Commodore 64 oltre 600 programmi di ogni genere: utility, word processor, bellissimi giochi. Andrea Polastri - Via S. Allende - 44100 Ferrara - Tel. 53194

Vendo oltre 1.500 giochi, ultimissime novità ancora non presenti in Italia; software gestionale, medico, professionale e moltissime utility. Prezzi bassissimi. Provare per credere. Vendo anche interfacce per collegare due registratori e duplicare cassette protette a L. 50.000. Franco Piccinno - Viale Leonardo da Vinci, 146 - 00145 Roma - Tel. 06/513215

Fantasmagorica offerta per VIC 20! Vendiamo giochi ed utility su cassetta a prezzi eccezionali (15 a L. 8.000 - 20 a L. 13.500 - 30 a L. 15.000). Inoltre possibilità di scegliere, fra quelli del nostro repertorio. Spedizioni in tutta Italia. Funzionamento della cassetta garantito. Per informazioni scrivere a: Stefano Bridi - Via S. Vito, 42 - 38050 Cognola (TN) - Tel. 0461/980115

Compro Commodore 64 con registratore entrambi in ottime condizioni a L. 400.000 trattabili. I programmi e i vari libri li compro a parte. Telefonare ore pasti e in serata. Giorgio Bollati - Via Molinetto di Lorenteggio, 15 - 20094 Corsico (MI) - Tel. 02/449161

Vendo, scambio videogiochi e software per VIC 20. Speditemi il vostro elenco richiedendomi la lista. Prezzi stracciati; annuncio sempre valido. Preferibilmente tratto con la zona di Bari. Pasquale Angiuli - Via Dante Alighieri, 270 - 70123 Bari - Tel. 217805

Vendo programma su cassetta per Commodore 64 e VIC 20 (16 Kbyte) per elaborare qualsiasi sistema Totocalcio, Enalotto, Totip, con condizionamento del numero dei segni e nei segni consecutivi a L. 50.000. Maurizio Caruso - Viale Libertà, 85 - 95014 Giarre (CT) - Tel. 932723

Vendo per Commodore 64 software a L. 3.000 al pezzo. Massima serietà. Scrivere o telefonare a: Borgheresi Simone - Via Montesanto, 17 - 53036 Poggibonsi (SI) - Tel. 0577/937336

Compro, compro, compro! Mandatemi la vostra lista ed io vi manderò la mia: cerco sopratutto cassette utility per il mio C64. Spedite tutto a: Saro Salimbene - Via Amendola, 36 - 98037 Letojanni (ME) - Tel. 0942/36276

Vendo VIC 20 più registratore C2N, joystick, libro "Il libro del Commodore VIC 20", cartridge "Super Alien", 50 programmi. Il tutto a L. 350.000. Marco Ghirardi - Via Anconetta, 30 - 32032 Feltre (BL)

Per CBM 64 cambio, vendo oltre 600 programmi di tutti i generi. Davide Bolognini - Via G. Salvemini, 14 - 61029 Urbino (PS) - Tel. 0722/38262

Vendo programma in LM di Mail List originale perchè scritto da me. Inoltre ottimo package per la gestione di archivi di libri, in LM, con ottime caratteristiche. Girano su CBM 64 ed Executive 64. Andrea Rigoni - Via G. Verdi, 16 - 35031 Abano Terme (PD) - Tel. 049/811447

Vendo o cambio vasta gamma programmi per CBM 64; giochi ed utility tra cui: Decathlon, Basket Ball NBA, Tab. Millesimali, programmi ingegneria ecc. Per eventuale cambio inviate la vostra lista. Renato Pietroniro - Via Lucera, 119 - 71100 Foggia - Tel. 0881/43373

Vendo giochi per CBM 64 su disco e cassetta. Scrivere a: Commodore Computer Club - Via Delle Albizzie, 40 - 00172 Roma

Vendo circa 1.000 programmi per Commodore 64 dai videogames eccezionali e utility con istruzioni. Scrivi subito o telefona a: Fausto Iori - Viale De Gasperi, 49 - 38023 Cles (TN) - Tel. 0463/21660

Vendo programmi bellissimi (giochi ed utility) a prezzi ottimi. Dispongo di: Magic Desk, Screen Graph, Pole Position, Zaxxon, Falcon Patrol, Frogger e tanti altri programmi per il vostro Commodore 64. Scrivete o telefonate a: Tony De Tora - Via Volturno, 27 - 81058 Vairano Patenorace (CE) - Tel. 0823/985397

Per CBM 64 vendo tutti i giochi e le utilities esistenti. E' indispensabile allegare francobollo da L. 450, se volete ricevere il catalogo. Massima serietà. Luigi Beviglia - Casella Postale 41 - 21052 Busto Arsizio

Compro programmi per CBM 64. Vendo le seguenti cassette per VCS 2600 Atari a metà prezzo: Pac Man, Spiderman, Pitfall, Oink, Decathlon, Enduro, Dragster, Tennis, Soccer, Raiders Of The Lost Ark, Donkey Kong, Keistone Kapers, Space Shuttle, Ghostman/Spikès Peak. Cristiano Cravanzola - Via Prof. Oliva, 16 - 12011 Borgo S. Dalmazzo (CN) - Tel. 0171/769659

Vendo Intellevision più 6 cassette quali Poker, Shark ark rog Bog, Space Battle, Space Armada, Triple Action al prezzo di L. 300.000. Danilo Savoca - Via L. Calatafimi, 8 - 90100 Palermo - Tel. 485618

Compro per VIC 20 espansione 16 Kbyte ad un prezzo molto vantaggioso. Telefonare o scrivere a: Osvaldo Bellomo - Via Della Croce - 75100 Matera - Tel. 0835/215430

Vendo o cambio giochi e programmi didattici per VIC 20. Per avere la mia lista inviare L. 450 in francobolli a: Brescianini Oscar - Via Veneto, 24 - 20031 Cesano Maderno (MI)

Cambio, vendo, compro programmi per VIC 20 versione base. Cerco espansione 3 Kbyte RAM. Mi interessano soprattutto i videogiochi. Giuliano Merlini - Viale Bovio, 131 - 64100 Teramo (TE) - Tel. 53088

Vendo ottimo corso di BASIC "Accademia" a L. 250.000 trattabili. Regalo all'acquirente il libro "Alla scoperta del VIC". G. Battista Gallus - Via S.Paolo, 3 - 09047 Selargius (CA) - Tel. 070/842795

Vendo qualsiasi programma per C64. Giochi in LM e BASIC - utility - didattici. Posseggo più di 500 programmi in cassetta. Telefonare per informazioni, scrivere per listino. Ogni gioco L. 3.000. Francesco Costantini - Via Zadei, 20 - 25015 Desenzano del Garda - Tel. 030/9142101

Vendo programmi di vario genere (più di 250), molti dei quali novità (es. Summer Games). Scrivere o telefonare ore pasti. Massima serietà. Carmine Parrella - Via T. Mommsen, 2 - 82100 Benevento - Tel. 0824/48068

Vendo programmi su cassetta per VIC 20 inespanso in LM: Scramble, Labirinto 3D, 7 e 1/2, Frogger, Flipper, Donkey Kong e molti altri, il tutto a L. 15.000. Per informazioni scrivere a: Fabrizio Giannotti - Via Giovanni XXIII, 18 - 88100 Catanzaro - Tel. 0961/29758

Per CBM 64 vendo, cambio programmi di qualsiasi genere, sia su disco che su nastro. Notevole quantità di software. Annuncio sempre valido. Dino Marocchi - Via Marconi, 302 - 65100 Pescara - Tel. 085/68352

Vendo programmi per C64 tutti in Turbo Tape. Alcuni esempi: Zaxxon, Pit Stop, Simon's BASIC; richiedere lista. Luigi Bettini - Via Venezia, 121 - 30039 Stra (VE)

SUPERCOMMODORE è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

....
....
....
....
....
....
....

Nome ....

Cognome ....

Via .... C.A.P. ....

Città .... Tel. ....

SUPERCOMMODORE

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soft - Viale Restelli, 5 - 20124 MILANO

INPUT/OUTPUT

☐ COMPRO ☐ VENDO ☐ CAMBIO

☐ VIC 20 ☐ C 16 ☐ C64 ☐ PERIF. ☐ SOFTWARE

....
....
....
....
....
....

Nome ....Cognome ....

Via .... C.A.P. ....

Città ....Tel. ....

SUPERCOMMODORE — LIBRI

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
J.soft – Viale Restelli, 5 - 20124 Milano - Tel. 6888228

Ordino i seguenti libri per un importo totale di L. ....+ L. 2.000
come contributo fisso per spese di spedizione

Cod. ....Cod. ....Cod. ....

Cod. ....Cod. ....Cod. ....

☐ Contanti allegati ☐ Assegno allegato n° ....

....

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul cc. n° 19445204 intestato a J.soft - Milano

☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Logo: potenza e semplicità	cod. AJAN026	L. 18.400
Linguaggio macchina del Commodore 64	cod. AJAN028	L. 27.100

Nome ....

Cognome ....

Via ....

Città .... C.A.P. .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i residenti all'estero: pagamento anticipato (vaglia o versamento su ns. ccp)



32 BIT 32 BIT

16 BIT 16 BIT

8 BIT 8 BIT



SINCLAIR QL: AL VERTICE DELLA NUOVA GENERAZIONE

Sinclair QL rivoluziona il mondo dei computer, perche combina le dimensioni di un home con la potenza e le capacita di un mini.
QL e l'unico computer, nella sua fascia, ad impiegare il microprocessore a 32 bit, quando gli altri si fermano a 8 oppure 16.
La sua portentosa memoria e di 128 KRAM espandibile a 640.
I quattro programmi applicativi, gia incorporati, sono immediatamente utilizzabili e superano, in qualità, il software dei microcomputer esistenti.

Ha la possibilità di multitask e puo essere inserito in reti di comunicazione.
Grazie ai due microdrive e al software incorporati, Sinclair QL, nella sua confezione originale, e gia pronto per l'uso: basta collegarlo ad un video.
E pensare che tutta questa tecnologia pesa meno di due chili e trova spazio in una normale 24 ore.
Un computer cosi non poteva che essere Sinclair.

sinclair

Distribuzione esclusiva: GBC Divisione Rebit.

Tutti i prodotti Sinclair, distribuiti da GBC Divisione Rebit, sono corredati da regolare certificato di garanzia italiana.